T.C.
MUĞLA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

BAYES YÖNTEMİ KULLANARAK İSTENMEYEN ELEKTRONİK POSTALARIN FİLTRELENMESİ

YÜKSEK LİSANS TEZİ

CÜNEYT ALTUNYAPRAK

MUĞLA 2006

T.C.
MUĞLA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

BAYES YÖNTEMİ KULLANARAK İSTENMEYEN ELEKTRONİK POSTALARIN FİLTRELENMESİ

YÜKSEK LİSANS TEZİ

CÜNEYT ALTUNYAPRAK

HAZİRAN 2006
MUĞLA

Yrd.Doç.Dr. Taner DİNÇER danışmanlığında Cüneyt Altunyaprak tarafından hazırlanan bu çalışma, 22/05/2006 tarihinde aşağıdaki jüri tarafından İstatistik ve Bilgisayar Bilimleri Anabilim Dalı'nda yüksek lisans tezi olarak oybirliği ile kabul edilmiştir.

Başkan Prof.Dr.Mübariz EMİNOV İmza :

Üye :Yrd.Doç.Dr.B.Taner DİNÇER İmza :

Üye :Yrd.Doç.Dr.İlhan Tarımer İmza :

ÖNSÖZ

Hızla büyüyen bir problem olarak karşımıza çıkan istenmeyen elektronik postaları engelleyebilmek için güvenilir filtreler geliştirmek bir ihtiyaç haline gelmiştir. Şimdiye kadar geliştirilen filtrelerin çoğu el ile geliştirilmiş anahtar kelimelere dayanan bir filtreleme gerçekleştirmişlerdir. Statik olarak niteleyebileceğimiz bu yöntem sürekli değişme eğiliminde olan istenmeyen elektronik postaların karakterlerini belirlemede ve onları filtrelemede yetersiz kalmaktadır.

Bu çalışmada, elektronik posta filtrelemede statik olarak kullanılan yöntemlere alternatif olarak son zamanlarda kullanımı hızla artan ve dinamik bir yaklaşım olan Bayes Filtreleme yöntemi tanıtılmış, incelenmiş ve bir uygulaması yapılarak elde edilen sonuçlar değerlendirilmiştir.

Çalışmamın geliştirilmesinde yardımcı olan ve yönlendiren hocalarım Yrd.Doç.Dr. Taner Dinçer, Dr. Mehmet Karahasan ve Muğla Üniversitesi Bilgi İşlem Daire Başkanı Osman Keleş'e teşekkürlerimi sunarım.

Cüneyt ALTUNYAPRAK

MUĞLA 2006

İÇİNDEKİLER DİZİNİ

ÖNSÖZ..... I

İÇİNDEKİLER DİZİNİ..... II

ÖZET..... V

ABSTRACT..... VI

ŞEKİLLER DİZİNİ..... VII

TABLOLAR/ÇİZELGELER DİZİNİ.....VIII

SEMBOLLER ve KISALTMALAR DİZİNİ..... IX

1. GİRİŞ..... 1

2. KAYNAK ÖZETLERİ..... 3

3. MATERYAL VE YÖNTEM..... 8

4. ELEKTRONİK POSTA.....10

4.1 Elektronik Posta Sistemi.....10

4.1.1 Elektronik posta istemcileri.....11

4.1.2 Elektronik posta sunucuları.....11

4.1.3 Elektronik posta mesajları.....14

4.2 İstenmeyen Elektronik Posta (Spam).....17

4.3 İstenmeyen Elektronik Posta Engelleme Teknikleri.....18

4.3.1 Kelime filtreleri.....18

4.3.2 Kural tabanlı puanlama sistemleri.....18

4.3.3 Bayes filtreleri.....19

4.3.4 Kara listeler 20

4.3.5 Gerçek zamanlı kara delik listeleri.....21

4.3.6 DNS Mx kayıt araması.....22

4.3.7 Ters DNS araması..... 22

4.3.8 Yeni ters DNS arama sistemleri.....23

4.3.9 Honeypots..... 24

4.3.9.1 Hashing sistemleri..... 24

4.3.9.2 Fingerprinting..... 24

4.3.10 Challenge/Response sistemleri.....25

4.3.11 Antivirüs taraması.....26

5.BAYES YAKLAŞIMI VE UYGULAMASI.....27

5.1 Belge Sınıflandırma 27

5.2 YönlendirmeliÖğrenme.....27

5.3 Yalın Bayes Sınıflandırma.....29

5.4 Bayes İstenmeyen Elektronik Posta Filtreleme.....34

5.5 Uygulama (Prototip Filtre)…….....38

5.5.1 Derlem oluşturulması…….38

5.5.2 Kelimelere ayırma…… 39

5.5.3 Kelime olasılıklarının hesaplanması …… .39

5.5.4 İstatistiksel kombinasyon……. .42

5.5.5 Prototip fitreleme programı tanıtımı ……...42

6. ARAŞTIRMA BULGULARI..... ...49

7. SONUÇLAR VE TARTIŞMA.....53

KAYNAKLAR 55

EK 1……. 57

ÖZGEÇMİŞ.....83

BAYES YÖNTEMİ KULLANARAK
İSTENMEYEN ELEKTRONİK POSTALARIN FİLTRELENMESİ

(Yüksek Lisans Tezi)

Cüneyt ALTUNYAPRAK

MUĞLA ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

2006

ÖZET

Bu çalışma, istenmeyen elektronik postaların otomatik olarak filtrelenmesi problemine Bayes yaklaşımının uygulanmasını incelemek ve performansı arttırabilecek ince ayarlar geliştirilebilmek amacıyla yapılmıştır. Öncelikle, Elektronik posta sistemleri, istenmeyen elektronik postaların karakteristikleri açıklanmış ve mevcut istenmeyen elektronik posta engelleme yöntemleri incelenmiştir. Daha sonra istenmeyen elektronik postaların engellenmesi bir otomatik öğrenme (belge sınıflandırma) işlemi olarak ele alınmış ve bu alanda yaygın olarak kullanılan Yalın Bayes Sınıflandırma tanıtılmıştır. Son olarak üretilen çözümler doğrultusunda Borland Delphi 7.0 yazılım geliştirme ortamında, Bayes yaklaşımı ile çalışan bir yazılım geliştirilmiştir. Çeşitli kullanıcılardan elde edilen istenmeyen ve normal elektronik posta mesajlarından oluşan derlemler üzerinde yazılım iki farklı olasılık modeli kullanılarak çalıştırılmış ve elde edilen sonuçlar etkinlik açısından karşılaştırılmıştır.

Kullanılan iki modelde de normal elektronik posta için Duyarlık %100 olmuştur. İstenmeyen elektronik postalar için de modellerde sırasıyla %81 ve %84 Duyarlık elde edilmiştir. Kelime oluş sayılarının da dikkate alındığı ikinci modelin filtreleme başarısını daha da arttırdığı gözlemlenmiştir.

Anahtar Kelimeler: BAYES, SPAM, İstenmeyen Elektronik Postaların Filtrelenmesi, Belge Sınıflandırma

Sayfa Adedi : 83

Tez yöneticisi : Yrd. Doç.Dr. Taner DİNÇER

FILTERING SPAM E-MAILS WITH BAYESIAN APPROACH

(M.Sc.Thesis)

Cüneyt ALTUNYAPRAK

MUĞLA UNIVERSITY

ISNTITUTE of SCIENCE and TECHNOLOGY

2006

ABSTRACT

This study aims to analyze Bayesian approach to the problem of filtering spam (junk) e-mails and to develop fine adjustments for increasing its performance. Firstly, electronic mail systems, characteristics of spam e-mails, and existing spam e-mail filtering techniques have been analyzed. Then, filtering spam e-mails problem was considered as a machine learning (document classification) task and a commonly used method for this domain that Naive Bayes Classification was introduced shortcomings and main problems in the existing systems and possible solutions to these are explained. Lastly, in accordance with the proposed solutions, a prototype filter working with Bayesian learning algorithm has been developed in Borland Delphi 7.0 software development environment. The prototype filter with two different probobalistic model is tested with a collection of spam (junk)e-mails and normal (regular) e-mails gathered from various users. The results have been compared for performance and accuracy.

In both models normal e-mails precision is %100. Spam e-mails precision for model 1 is %81 and model 2 is %84. Our secondary model which considers the word occurring performed far better than the first model.

Key Words : BAYES, SPAM, Filtering Spam E-mail, Document Classification

Page Number : 83

Advisor : Yrd. Doç.Dr. Taner DİNÇER

ŞEKİLLER DİZİNİ

Şekil No Sayfa No

Şekil 1.1 Yalın Bayes sınıflandırma formülü….....32

Şekil 2.1 Filtre için kelime veritabanı yaratılması.....35

Şekil 3.1 Filtre ana menü.....42

Şekil 3.2 Filtre veritabanı ekranı …….…….…….....43

Şekil 3.3 Filtre olasılık hesaplama ekranı …….……....44

Şekil 3.4 Filtre ayarlar ekranı …….…….……..45

Şekil 3.5 Filtre istatistik ekranı …….…….....45

Şekil 3.6 Filtre öğrenme ekranı …….……...46

Şekil 3.7 Filtre sınıflandırma ekranı**..**…….…….……...47

Şekil 3.8 Filtre analiz ekranı …….…….……48

TABLOLAR/ÇİZELGELER DİZİNİ

Şekil No Sayfa No

Tablo 1.1 Derlem..... 9

Tablo 1.2 Yazılım geliştirme ortamı özellikleri 9

Tablo 2.1 Örnek veri kümesi 31

Tablo 2.2 Frekanslar ……..... 31

Tablo 2.3 Olasılıklar..... 32

Tablo 3.1 Olasılık tablosu.....49

Tablo 3.2 Karşılaştırmalı olasılık tablosu…….....51

SEMBOLLER ve KISALTMALAR DİZİNİ

ADO	Active Data Objects
AI	Artifical Intelligence
AOL	American On Line
ASCII	American Standart Code of Information
DMP	Designated Mailers Protocol
DNS	Domain Name Service
FTP	File Transfer Protocol
IMAP	Internet Message Access Protocol
IR	Information Retrieval
ISP	Internet Service Provider
MIME	Multipurpose Internet Mail Extensions
ML	Machine Learning
MTA	Mail Transfer Agent
MX	Mail Exchanger
NB	Naive Bayes
NBC	Naive Bayes Classifier
NLP	Naturel Language Processing
POP	Post Office Protocol
POP3	Post Office Protocol 3
RBL	Real Time Black Hole List
RFC	Request For Comments
RMX	Reverse Mail Exchanger
SMTP	Simple Mail Trasfer Protocol
SPF	Sender Permitted From
SVM	Support Vector Machines
TCP/IP	Transmission Control Protocol/ Internet Protocol

1. GİRİŞ

Bilgisayar kullanımının yaygınlaşması ve internetin hızla gelişmesi sonucunda elektronik posta hayatımızda giderek önemi artan bir haberleşme ortamı haline gelmiştir. Çok hızlı ve en ekonomik haberleşme biçimi olduğu için kullanıcılarının sayısı da kısa zamanda büyük ölçüde artmıştır. Elektronik posta, sadece iki insan arasındaki iletişimi sağlayan bir ortam olarak değil, elektronik ticaret yönetimi için de iyi bir ortam olarak popüler olmuştur.

Bir taraftan internet üzerinde ürünlerinin (çoğunlukla güvenilmeyen, hileli ürünler) ticari reklâmını yapmak veya yasal olmayan duyurularda bulunmak isteyenler, diğer taraftan da elektronik posta adres listelerinin sayısının artması geçtiğimiz birkaç yıl içerisinde "istenmeyen elektronik posta" (spam) kavramının ortaya çıkmasına sebep olmuştur. Bugün çok ileri boyutlara ulaşan ve kullanıcılar açısından zaman kaybı, işletmeler açısından ise verimlilik kaybına sebep olan istenmeyen elektronik posta probleminin önüne geçebilmek amacıyla bu tür elektronik postaları otomatik olarak filtreleyebilen metotlar bir gereklilik haline gelmiştir. İstenmeyen elektronik posta problemini tamamen çözebilmiş bir teknik veya tekniklerin birleşmesinden oluşan bir çözüm mevcut değildir. İstenmeyen elektronik posta trafiği çok yoğun kuruluşlar için yapılacak en iyi şey birkaç tekniği birleştirmek olacaktır. Bu tekniklerin en yenilerinden birisi olan Bayes Filtreleme artık istenmeyen elektronik posta engelleme yazılımlarının bileşenleri arasında son derece başarılı sonuçlar üreterek yerini almıştır.

Bayes Filtreleme, teoride metin sınıflandırma metotlarından Yalın Bayes sınıflandırmanın elektronik postalara uygulanmasıyla geliştirilmiştir. Günümüzde pratik olarak farklı yöntem ve yaklaşımların uygulandığı ve başarısını ispat etmiş bir teknik olmuştur.

Bu çalışmada istenmeyen elektronik postaların filtrelenmesi bir belge (metin) sınıflandırma problemi bağlamında ele alınmıştır. Geliştirilen prototip yazılımla Bayes Filtreleme Türkçe istenmeyen ve normal elektronik posta mesajlarından oluşturulmuş bir derleme farklı yollarla uygulanmış ve performans değerlendirmesi, karşılaştırması yapılmıştır.

Tez akışı içerisinde öncelikle elektronik posta sisteminin genel yapısı ve mevcut istenmeyen elektronik posta engelleme teknikleri tanıtılmıştır. Materyal ve Yöntem bölümünde geliştirilen prototip Bayes filtresinin işleyişi, menüleri ve kullanımı hakkında bilgi verilmiştir.

Beşinci bölümde de belge sınıflandırma, yönlendirmeli öğrenme kavramları ve Yalın Bayes sınıflandırmanın istenmeyen elektronik posta problemine uygulanması anlatılmıştır.

2. KAYNAK ÖZETLERİ

Bir istenmeyen elektronik posta filtreleme sistemi klasik elektronik posta istemcilerindeki gibi isme, konuya veya gönderene göre statik filtreleme yapan bir uygulamadan daha güçlü olmalıdır. Elektronik postayı verimli bir şekilde sınıflandırabilmek için filtre her bir mesajın bütün metnini analiz etmek zorundadır. Ayrıca filtre kendi dinamik elektronik posta ortamındaki değişikliklere kolayca uyum sağlayabilmelidir.

Son yapılan teorik araştırmalar, belge sınıflandırma (metin sınıflandırma) problemi etrafında yoğunlaşmıştır. Çoğu yayınlar sorunları bu problemle tanımlar. Diğer taraftan, istenmeyen elektronik postaların filtrelenmesi veya sınıflandırması bilimsel araştırmalar için yeni bir konudur. İstenmeyen elektronik postaların otomatik olarak filtrelenmesini tanımlayan çoğu yayın, metin sınıflandırmanın istenmeyen elektronik postaların otomatik olarak filtrelenmesi problemine uygulanabilirliğine dair teorik konularla ilgilidir. Çeşitli otomatik öğrenme algoritmaları bu probleme uygulanmış, daha önceden hazırlanmış herkesin kullanımına açık derlemler üzerinde çalışma sonuçları karşılaştırılmıştır. Bu öğrenme algoritmaları arasında Yalın Bayes algoritmasının uygulamadaki kolaylığı ve elde ettiği başarılar dikkat çekmektedir. Bununla birlikte Yalın Bayes sınıflandırma bağlamında çeşitli istatistiksel modellerin karşılaştırılması da yapılmıştır.

Bu kısımda, çalışmada yararlanılan kaynaklar tanıtılmıştır. Bu çalışmaların metot ve sonuçları kısaca açıklanmıştır.

Cohen (1996)'in çalışmasında elektronik posta sınıflandırma ve filtreleme problemine odaklanılmış ve (learning text classifiers) öğrenen metin sınıflandırıcıları için TF-IDF (weighting) ağırlıklandırmaya dayanan bir geleneksel bilgi geri getirim metoduyla, kural tabanlı bir öğrenme algoritması olan RIPPER'in (Cohen 1995a) genişletilmiş bir versiyonu karşılaştırılmıştır. Genişletilmiş RIPPER algoritması çeşitli derlemler üzerinde en iyi sonucu vermiştir. İki metodunda büyük talim kümeleriyle bile iyi çalıştığı, az sayıda örnekten anlamlı genellemeler elde ettiği, elektronik posta sınıflandırma

problemlerinde performans açısından karşılaştırılabilir olduğu ispat edilmiştir. Yapılan deneylerde genişletilmiş RIPPER algoritması en iyi sonuçları vermiş bununla birlikte TF-IDF (Salton 1991) de iyi performans göstermiştir.

Sahami (1998)'nin çalışmasında, istenmeyen elektronik postaların filtrelenmesi, sadece mesajın ham metnine dayanan bir metin sınıflandırma problemi olarak değerlendirilmemiş, elektronik posta alanına özgü bazı özellikler de Bayes sınıflandırıcı ile birleştirilmiştir. 1578 tanesi "istenmeyen" ve 211 tanesi "normal" olarak önceden sınıflandırılmış, 1789 adet gerçek elektronik posta mesajından oluşan bir derlem üzerinde bir dizi deney yapılmıştır. Öncelikle özellik kümesi olarak sadece mesajların konu ve gövdelerinde geçen kelime esaslı simge (word-based token) göz önüne alınmıştır. Daha sonra 35 adet elle hazırlanmış (phrasal features) deyimsel özellik eklenmiş ve son olarak da 20 tane metinsel olmayan alana özgü özellik ile, özellik kümesi geliştirilmiştir. Yalnızca kelimelere dayalı bir Bayes sınıflandırmaya sırasıyla özel ifadeler (örneğin : "Free Money", "FREE!", "only $" v.s) ve alana özgü özellikler (örneğin mesajı gönderenin alan adı tipi; edu,com, v.s) eklendiğinde performans artmış, başarılı sonuçlar elde edilmiştir. Sadece kelime esaslı simgelere dayanan sınıflandırmada istenmeyen e-posta için duyarlık %97.1, anma %94.3 olurken normal elektronik posta için duyarlık %87.7, anma %93.4 olmuştur. Özellik kümesini kelime esaslı simgeler ve özel ifadelerin oluşturduğu sınıflandırmada istenmeyen elektronik posta için duyarlık %97.6, anma %94.3 olurken normal elektronik posta için duyarlık %87.8, anma %94.7 olmuştur. Özellik kümesini kelime esaslı simgeler, özel ifadeler ve alana özgü özelliklerin oluşturduğu sınıflandırmada ise sınıflandırmada istenmeyen elektronik posta için duyarlık %100.0, anma %98.3 olurken normal elektronik posta için duyarlık %96.2, anma %100.0 olmuştur.

McCallum ve Nigam (1998)'ın çalışmasında, en son metin sınıflandırma yaklaşımlarının, sınıflandırma için ikisi de Bayes varsayımı yapan iki farklı birinci dereceden (first-order) olasılıksal model kullandığı belirtilmiştir. Bazıları, kelimeler ve (binary) ikili kelime özellikleri arasında bağımlılığın olmadığı bir Bayes ağı (network) olan Multi-variate Bernoulli model kullanmışlardır. Diğerleri ise, kelime sayıları ile uni-gram dil modeli olan Multi-nomial model kullanmışlardır. Bu çalışmada bu iki modelin farklılıkları ve detayları açıklanmış

ve sınıflandırma performansları karşılaştırılmıştır. Sonuçta Multi-variate Bernoulli model küçük kelime dağarcıklarıyla iyi performans göstermiş, fakat multi-nomial model büyük kelime dağarcıklarıyla bile çoğunlukla daha iyi performans sergilemiştir. Deneyler sonucunda Multi-nomial modelin hatayı ortalama %27 bazen de %50 oranında azalttığını görmüşler.

Carreras (2001)'ın çalışmasında, istenmeyen elektronik postaların otomatik olarak filtrelenmesinde bir dizi karşılaştırmalı deney yapılmış, Ada Boost algoritması probleme uygulanmıştır. Ücretsiz olarak herkesin kullanımına açık olan PU1 derlemi üzerinde, Boosting (boosting-based) esaslı metotlar temel öğrenme algoritmalarından Yalın(Naive) Bayes ve Induction of Decision Trees 'i çok yüksek seviyede F1 ölçüleri elde ederek performans bakımından geçmişlerdir. Temel öğrenicilerin karmaşıklığını arttırmanın daha yüksek kesinlikte (yani istenmeyen ve normal elektronik postaların hatalı sınıflandırılmasına daha az müsaade eden bir senaryo) sınıflandırıcılar elde etmeyi mümkün kıldığı sonucuna da varılmıştır.

Z. Chuan, L. Xianliang, H. Mengshu, Z. Xu (2005)'nun çalışması, ANN-LVQ yaklaşımı istenmeyen elektronik postaları tanımak için sırayla istenmeyen elektronik postayı içeriğine göre (online alışveriş, politik içerik, pornografi vs.) çeşitli alt sınıflara ayırır. Sonra istenmeyen elektronik postayı tanımak için bu alt sınıflar LVQ yapay sinir ağıyla karmaşık sınıflara birleştirir. Bu çalışmada http://www.spamassassin.org/publiccorpus web adresinde bulunan derlemden rasgele seçilmiş 580 istenmeyen elektronik posta ve 420 normal elektronik postadan oluşan bir derlem üzerinde çalışılmıştır. Tamamen İngilizce e-postaların yer aldığı bu derlemde elektronik postaların ekleri ve konu üstbilgisi dışındaki e-posta üstbilgileri çıkarılmıştır.Yapılan deneylerde yapay sinir ağının talim sayısı filtrenin performansını etkilemiştir. Yeterli talim yapılamadığında filtrenin performansının ideal olmadığı gözlemlenmiştir. Karşılaştırılan iki yapay sinir ağı algoritmasının da Bayes esaslı algoritmadan genellikle daha iyi olduğu gözlenmiş, bunun sebebinin de Yapay Sinir Ağlarının, bütündeki her bir özellik kelimesi arasındaki ilişki avantajını kullanması, Bayes esaslı algoritmanın ise basitçe özellik kelimelerini bağımsız olarak ele alması olduğu ileri sürülmüştür.

Cormac O'Brien ve Carl Vogel (2003)'in çalışmasında, istenmeyen elektronik postayı tanımada iki istatistiksel metot karşılaştırılmış, problemin büyüklüğü tartışılmıştır. Bayes metodu "chi by degrees of freedom" yaklaşımıyla karşılaştırılmış, iki metot da ümit verici sonuçlar sağlamıştır. Ancak "chi by degreess of freedom" yanlış alarmları (false positive) düşürmeye yardımcı olan anlamlılık ölçüleri sağlama avantajına sahip olduğu ve karakter seviyeli simgelemenin (tokenization) kelime seviyeliye göre daha etkin olduğu gösterilmiştir.

L. Zhang, J. Zhu ve T. Yao (2004)'nun çalışmasında istatistiksel olarak istenmeyen elektronik posta filtreleme bağlamında beş tane yönlendirmeli öğrenme metodu karşılaştırmıştır. (Bayes, Maximum Entropy, Ada Boost, SVM, Memory-based learning). Özellik seçiminin anlamlılığının sınıflandırıcıdan sınıflandırıcıya çok büyük çeşitlilikler gösterdiği gözlemlenmiştir. "Bag of Features" filtreleme modelinin istenmeyen elektronik posta filtreleme işinde oldukça etkin olduğu, SVM, Maximum Entropy ve Ada Boost 'un cost-sensitive istenmeyen elektronik posta filtreleme işinde en iyi performansı gösterdiği ve mesaj başlıklarındaki bilginin mesaj gövdesindeki bilgi kadar önemli olduğu sonuçlarına varılmıştır.

J. Rennie (2000)'nin çalışması, metin sınıflandırmanın elektronik posta filtreleme problemine uygulanmasına dair pratik bir çalışmadır. Yazar "ifile" adında Yalın(Naive) Bayes algoritmasıyla çalışan ve EXMH posta istemcisine adapte edilebilen bir filtre geliştirmiş ve yaptığı deneylerden bu yazılımın faydalı olabileceği sonucuna varmıştır.

Y. Diao, H. Lu ve D. Wu. (2000)'nun çalışmasında bir Naive Bayes sınıflandırıcı, Decision Tree tabanlı bir sınıflandırıcı ile karşılaştırılmış; iki sınıflandırıcının da filtrelemeyi makul bir doğrulukla gerçekleştirdiği gözlemlenmiştir. Özellik ve talim boyutları ikisi için de uygun olarak seçildiğinde Decision Tree tabanlı sınıflandırıcı, Bayes tabanlı sınıflandırıcıyı performans olarak geçmiş ve dikkatlice hazırlanmış bir Bayes sınıflandırıcının daha sağlam olduğu iddia edilmiştir.

Androutsopoulos (2000)'un çalışmasında, istenmeyen elektronik posta filtreleme bağlamında iki tane ML algoritmasının performansı incelenmiştir.

Uygun cost-sensitive değerlendirme ölçüleri tanıtıldıktan sonra Naive Bayes filtresinin performansı alternatif bir memory-based öğrenme yaklaşımı ile karşılaştırılmıştır. İki metot da istenmeyen elektronik posta filtrelemede yüksek doğruluk sağlamış ve yaygın olarak kullanılan bir e-posta istemcisinin anahtar kelime tabanlı filtrelemesini performans olarak açıkça geçmişlerdir. Çalışma sonucunda, learning based antispam filtrelerinin yapılması, istenmeyen elektronik posta mesajları kolayca işaretlendiğinde veya bloke edilen mesajları gönderenleri bildirmek için ek mekanizmalar mümkün olduğunda tamamıyla uygun olduğu önerisi yapılmıştır. Böyle bir mekanizmanın olmadığı durumlarda memory-based yaklaşımın daha uygulanabilir olduğu öne sürülmüştür.

J. Provost (1999), "Bag valued" özellikleriyle Naive Bayes ve Ripper kural tabanlı öğrenme algoritmasını e-posta sınıflandırma ve istenmeyen elektronik posta filtreleme işlerinde karşılaştırmıştır. Deneyler sonucunda Naive Bayes Ripper'i sınıflandırma doğruluğunda performans olarak geride bırakmıştır.

H. Drucker (1999), elektronik postayı istenmeyen elektronik posta veya normal elektronik posta olarak sınıflandırmada Support Vector Machine algoritmasını kullanmış ve sonuçları diğer üç sınıflandırma algoritması Ripper, Rocchio, Boosting Decision Trees ile karşılaştırmıştır. Boosting Trees ve SVM doğruluk ve hızda kabul edilebilir test performansı göstermişler. Ancak SVM'nin talim zamanı anlamlı düzeyde daha az olmuştur. Boosting daha az hata oranına sahipken hataların dağılımında SVM daha iyi çıkmıştır. Stop List kullanılıp kullanılmaması seçiminde Stop List kullanılmamasını tercih edilebilir bulmuşlardır.

3. MATERYAL VE YÖNTEM

İstenmeyen elektronik postaların filtrelenmesi bir belge(metin) sınıflandırma problemi olarak ele alınmış ve belge sınıflandırma algoritmalarından Yalın Bayes algoritması altında ele alınabilecek Bayes Filtreleme tekniği, geliştirilen bir prototip yazılımla, çeşitli kullanıcılardan toplanmış istenmeyen ve normal elektronik posta mesajlarından oluşan bir derlem üzerinde test edilmiştir. Bu işlem yapılırken derlem, talim kümesi ve test kümesi olarak iki kısma ayrılmıştır. İki farklı olasılık modeli kullanılarak geliştirilen prototip filtre önce talim kümesi üzerinde öğrenme işlemini gerçekleştirmiş sonra da test kümesi üzerinde filtreleme işlemi yapmıştır. Elde edilen sonuçların etkinliği ölçme ve değerlendirme yöntemlerinden Duyarlık ve Anma ile değerlendirilmiştir.

Bayes Filtreleme, Windows XP işletim sistemi üzerinde Borland Delphi 7.0 programlama dili kullanılarak geliştirilmiş prototip yazılımla probleme uygulanmıştır. Derlemler, kullanıcıların elektronik posta istemcilerinden (Microsoft Outlook ve Microsoft Outlook Express) Microsoft Access programına aktarılarak prototip yazılıma veritabanı olarak tanıtılmış ve ADO bağlantısıyla işlemler gerçekleştirilmiştir.

Filtreleme işlemi gerçekleştirilirken izlenen adımlar ayrıntılı olarak açıklanmıştır. Geliştirilen prototip ile farklı derlemlerde Bayes Filtreleme işleminin kolaylıkla gerçekleştirilebilmesi ve sonuçların karşılaştırılması amaçlanmıştır.

Kullanılan derlemlerde birbirinin eşi olmayan, tamamı Türkçe içerikli elektronik posta mesajları kullanılmıştır. Test kümesinde kullanılan elektronik posta mesajlarının tamamının alınma tarihleri, talim kümesinde (öğrenmede) kullanılan elektronik posta mesajlarının alınma tarihlerinden sonradır. Bu filtrenin başarımının ölçülmesinde son derece önemli bir noktadır. Ayrıca derlemde kullanılan istenmeyen elektronik posta mesajlarının büyük çoğunluğu birbirine çok fazla benzemeyen ve farklı kullanıcıların posta kutusuna gelen mesajlardan seçilmiştir. Normal elektronik posta mesajları ise tek bir kullanıcının

elektronik posta kutusundan derlenmiştir. Daha sonra bu hususların sonuçlar üzerine etkisi tartışılmıştır. Elektronik posta mesajları kelimelere ayrılırken bu işlem sonunda ortaya çıkan kelimelerin anlamlı olup olmadığı incelenmemiştir. Örneğin "aslında" kelimesinin "aslinda" olarak yazılması gibi durumlara dikkat edilmemiş, bunlardan her biri ayrı bir kelime olarak kullanılmıştır. Herhangi bir dilbilimsel yaklaşım kullanılmamıştır. Bu şekilde elektronik posta mesajlarından ayıklanan kelimelere gösterge demek daha doğru olacaktır. Çalışmada kelimelere ayırmadan kastedilen bu işlemdir. Tablo 1.1 Kullanılan derlemin detaylarını ve Tablo 2.2 de filtrenin geliştirildiği yazılım geliştirme ortamının özeliklerini göstermektedir.

Tablo 1.1 Derlem

Talim Kümesi	Elektronik Posta	Kelime olarak ayrılan gösterge
İstenmeyen elektronik posta	667	21254
Normal elektronik posta	1420	48549
Toplam	2387	60165

Toplam kelime olarak ayrılan gösterge sayısı ortak göstergeler (kelimeler) olduğu için her iki sınıftan ayrılanların doğrudan toplamından ortak kelimeler çıkarılarak hesaplanmıştır.

Tablo 1.2 Yazılım geliştirme ortamı özellikleri

İşletim Sistemi	Windows XP
Programlama Dili	Borland Delphi 7.0
Hafıza	1 GB RAM
İşlemci	2.0 GHZ Intel Pentium M CPU

4.ELEKTRONİK POSTA

4.1 Elektronik Posta Sistemi

Elektronik posta (e-posta) bir kullanıcının bir bilgisayar sisteminde yazdığı, onu okuyabilecek başka bir kullanıcıya bir çeşit bilgisayar ağı üzerinden ilettiği, genellikle basit bir metin mesajı şeklinde olan elektronik mesaja verilen addır. Elektronik posta mesajları mektuplara benzerler ve başlıca iki ana kısımdan oluşurlar. Üstbilgi (header) alıcının, mesajın kopyasının gönderileceği kişinin adını ve adresini ve mesajın konusunu içerir. Gövde (body) mesajı içeren kısımdır. Bir mektubu yollarken olduğu gibi elektronik postayı gönderirken de doğru adrese ihtiyaç vardır. Eğer yanlış veya eksik bir adres yazarsanız, mesajınız size alıcıya ulaşılamadı böyle bir alıcı yok v.s gibi bir konuyla size geri döner. Bir elektronik posta mesajı aldığınızda, üstbilgiler size mesajın nereden geldiğini, ne zaman ve nasıl gönderildiğini belirtir.

Mektupları bir zarfla kapatırız fakat elektronik posta mesajlarında böyle bir durum yoktur, elektronik posta mesajı özel değildir. Daha çok bir posta kartına benzerler. Mesajların onlara bakması gerekli olmayan kişiler tarafından yolu kesilebilir ve okunabilirler. Eğer şifreleme kullanılmazsa, gizli bilgilerin elektronik postayla gönderilmesinden kaçınılmalıdır.

İnternet üzerinde veya bir bilgisayar ağı içerisinde elektronik posta göndermemizi sağlayan protokol, TCP/IP protokol kümesinin en üst kısmı olan uygulama katmanındaki SMTP protokolüdür. POP3 veya IMAP gibi protokoller sayesinde ise posta kutumuza ulaşan mesajlara erişebiliriz.

4.1.1 Elektronik posta istemcileri

Elektronik postaları okumak için Microsoft Outlook, Microsoft Outlook Express, Eudora veya Pine gibi bir yazılım kullanılır. Eğer Hotmail, Yahoo veya Gmail gibi ücretsiz bir elektronik posta servisine üye olunursa, elektronik postaları okumak için bir web sayfası olarak görünen bir elektronik posta istemcisi kullanılır.

Bütün elektronik posta istemcilerinin 4 temel işlevi vardır:

1-) Posta kutunuzdaki mesajların bir listesini görüntülemek,

2-) Listeden bir mesaj seçmek ve mesajın içeriğini okumak,

3-) Yeni bir mesaj oluşturmak ve göndermek,

4-) Eklentileri işlemek - göndereceğiniz bir mesaja eklenti yerleştirmek veya alınan bir mesajdaki eklentileri kaydetmek.

4.1.2 Elektronik posta sunucuları

Elektronik posta göndermek ve almak için, istemcinin bağlanması ve elektronik posta mesajını teslim edilmek üzere vermesi için internet üzerinde özel bir bilgisayara ihtiyaç vardır. Bu bilgisayar elektronik posta servisi sağlamak için bir uygulama yazılımı çalıştırır ve elektronik posta sunucu olarak adlandırılır.

İnternet üzerinde Web sunucu, Ftp sunucu, Telnet sunucu, Elektronik Posta sunucu v.s olarak uygulama yazılımları çalıştıran milyonlarca bilgisayar vardır. Bu uygulamalar, sunucu makine üzerinde sürekli çalışır ve özel portları dinlerler. İnsanların veya programların bağlanması için beklerler. SMTP sunucusu giden ve gelen elektronik postaları işler. Bu sunucu elektronik posta göndermek isteyen birisi için 25 numaralı portu dinler. Elektronik posta istemcisi posta göndermek için SMTP sunucu ile bir etkileşimde bulunur.

Şimdi bir örnekle bu etkileşimin nasıl olduğunu gösterelim.

1-) Muğla Üniversitesi'nde bir hocanın, bilgisayarını açtığı ve bir öğrencisine şu şekilde bir elektronik posta mesajı yazdığı varsayılsın. Elektronik posta istemcisi olarak da Pine kullandığı varsayılsın.

Değerli öğrencim , Projene A verdim.

İyi çalışmalar ……

2-) Hoca, elektronik posta adresi hasanyildirim@hotmail.com olan öğrencisine mesaj göndermek için e-posta istemcisinde gönder düğmesine tıklar.

3-) Elektronik posta istemcisi Pine, Muğla Üniversitesi SMTP sunucusuna 25 numaralı portu kullanarak bağlanır.

4-) Pine gönderenin adresini, alıcının adresini ve mesajın gövdesini SMTP sunucusuna verir.

5-) SMTP sunucusu "to" , "kime" (hasanyildirim@hotmail.com) adresini alır ve iki parçaya ayırır.

A-) alıcı adı (hasanyildirim)

B-)domain adı (hotmail.com)

Eğer hoca elektronik postayı öğrencinin Muğla Üniversitesi'ndeki hesabına göndermişse, SMTP sunucu elektronik postayı basitçe Muğla üniversitesinde gelen maili işleyen sunucuya verecektir. Gelen elektronik postalar POP veya IMAP sunucuları aracılığıyla işlenir.

6-) Alıcı farklı bir domain de olduğu için, SMTP'nin bu domain le haberleşmesi gerekir. SMTP sunucusu, DNS sunucusuna hotmail.com için SMTP sunucusunun ip adresini bulmak için bir sorgu gönderir. DNS sunucusu hotmail için tanımlı olan bir veya daha fazla SMTP sunucusunun ip adresiyle bu sorguyu yanıtlar (MX record).

7-) Muğla Üniversitesi'ndeki SMTP sunucusu 25 numaralı portu kullanarak Hotmail'in SMTP sunucularından birisine bağlanır. Diğer SMTP ile aralarında çok basit bir metin diyalogu geçer ve mesajı ona aktarır. Hotmail

sunucusu hasanyildirim için domain adının Hotmail'de olduğunu onaylar ve mesajı, hasanyildirimin posta kutusuna yerleştiren, Hotmail'in IMAP sunucusuna aktarır.

SMTP RFC 821 ve RFC 1123 de tanımlanmıştır. 7-bit US-ASCII karakterinde düzenli bir veri akışı talep eder, gönderenin SMTP komutlarını alıcıya yayınlamasıyla diyalog başlar (SMTP SESSION). Alıcı göndereni, ardından kod hakkında ek bilginin yer aldığı, sayısal yanıt kodlarıyla yanıtlar.

SMTP sunucu, HELO, MAIL, RCPT, ve DATA gibi çok basit metin komutlarını anlar. En yaygın kullanılan komutlar şunlardır:

- HELO : Sunucu kendisini tanıtır.

- EHLO : Sunucu kendisini tanıtır ve genişletilmiş modu talep eder.

-MAIL FROM : Göndereni tanımlar

-RCPT TO : Alıcıyı tanımlar

-DATA : Mesajın gövdesini tanımlar.

-RSET : Bağlantıyı yeniden başlatır.

-QUIT : Oturumu sonlandırır.

Aşağıdakiler alıcı SMTP tarafından gönderilen bazı yanıt kodlarıdır.

• 211 System Status or system help reply

• 220 domain Service ready

• 221 domain Service closing transmission channel

• 250 Requested action OK and completed

• 354 Start mail input; end with .

• 421 Domain service not available, closing connection

• 450 Mailbox unavailable, requested mail action not taken

Tipik bir değişimde, gönderen alıcı ile bağlantı kurduktan sonra, alıcı hazır olduğunu gösteren 220 koduyla yanıt gönderir. Gönderen daha sonra HELO

komutunu bir argüman olarak gönderir. HELO komutu göndereni alıcıya tanıtır, ve alıcı sonra 250 yanıt koduyla yanıtlar. Bu gönderene bağlantının açık olduğunu ve devam etmek için hazır olduğunu söyler. Bir sonraki adım gönderenin ve alıcının sunucu adresini tanımlar ve doğrular. Kendisini tanıttıktan sonra, elektronik posta istemcisi "from" ve "to" adreslerini belirtir ve sonra alıcının mesajı almaya hazır olup olmadığını sormak için DATA komutunu yayınlar. Alıcı gönderenin mesajı teslim edebileceğini belirten 354 koduyla yanıt verir. İletim bir satırda yalnız bir '.' ile biter. .mesaj gönderildikten sonra, gönderen "quit" komutunu yayınlar ve alıcıda 221 yanıt koduyla yanıtlar. Aslında bu diyalog bir SMTP sunucusunun 25 numaralı portuna telnet ile bağlanılarak gerçekleştirilebilir. Dolayısıyla böyle bir işlem sonunda spoof (aldatma) yapılabilmektedir.

4.1.3 Elektronik posta mesajları

Elektronik Posta mesajları herhangi bir lokal bilgisayar ağı veya internet üzerindeki bilgisayarlar arasında SMTP kullanılarak iletilirler. Elektronik posta mesajı iletilirken gönderen SMTP, gönderenin adresini, alıcıyı ve gönderilecek veriyi belirler. Veri isteğe bağlı bir ASCII metindir. ASCII, bilgisayarlar arasında metin aktarımı sırasında karakterleri sayısal olarak temsil etmek için kullanılan standart 7 bitlik koddur. SMTP herhangi bir şekilde veriyi yorumlamaz veya veri için bir format belirlemez.

Elektronik postanın yeni kullanılmaya başladığı zamanlarda, veri içinde konu, mesajın gönderildiği tarih gibi bilgileri içeren bir üstbilgiye ihtiyaç duyulmuştur. Sonuç olarak kişisel olarak geliştirilmiş, uyumsuzluğa yol açan, resmi olmayan birçok standart geliştirilmiştir. Bu özelliklerin birbirine yakın olmasını sağlama ve uygulamaları düzenleme ihtiyacı doğmuş; bu durum, hemen arkasından mesaj içeriğinin yer aldığı mesaj üstbilgilerinin bir standart kümesini belirleyen ve daha sonradan RFC 822 (Standart for the format of arpa internet text messages) ile güncellenen RFC 733'de sonuçlandırılmıştır. RFC 822 ile ilgili

problem mesaj içeriğinin sadece ASCII metin olmasına izin vermesidir. MIME (çok amaçlı internet posta uzantıları) bu sınırlamanın üstesinden gelmiş ve ASCII den farklı karakter kümeleri taşıyan, metin olmayan veri(ekler) ve çok parçalı mesajlar gibi mesajlara olanak sağlamıştır.

Bir elektronik posta mesajı üstbilgi ve gövde olarak iki kısımdan oluşur. Üstbilgi elektronik posta hakkında gönderenin adresi, alıcının adresi ve iletim tarihi gibi bazı bilgileri içerir. Bu bilgi Anahtar:Değer çifti olarak özel bir formatta RFC 822 de tanılanmıştır. Örneğin Return-Path : cuneyt@mu.edu.tr. Gövde kısmında ise gönderilecek olan mesaj ve ekler yer alır.

Date, From ve To (veya In-Reply-To, BCC) gibi bazı üstbilgiler zorunludur. Subject, CC, Received ve Message-Id gibi diğerleri ise isteğe bağlıdır, fakat sıkça kullanılırlar. Çoğu üstbilgiler elektronik posta sistemi tarafından göz ardı edilir. Fakat bütün üstbilgiler alıcının sunucuna tanınmasalar bile aktarılırlar. Örneğin X ile başlayan üstbilgiler kişisel uygulamaları veya kurumsal kullanım içindir.

Received üstbilgi alanları önemlidir, çünkü mesajın izlenmesini mümkün kılar. Bir mesaj planlanmış alıcısına giderken sunucular üstbilgiye ek bir Received satırı eklerler. Bu satırlar üstbilgilerin en üstüne eklenir, böylece mesajı alan ilk sunucu Received satırlarının en altında yer alır. İstenmeyen elektronik postacılar elektronik postalarına sıklıkla sahte Received satırları eklerler.

Elektronik postanın gövdesi üstbilgilerden boş bir satırla ayrılır. Bir elektronik postaya bir eklenti iliştirildiğinde, eklenti mesajın gövdesinde yer alır. Eklentiler olsa bile elektronik posta mesajı halen sadece metin mesajıdır.

MIME olarak adlandırılan bir internet standardı, mesajın nasıl biçimlendirileceğini ve eklentilerin mesajdan nasıl ayrılacağını belirler. MIME kodlaması hakkındaki bilgi, RFC de belirtilen üstbilgi alanlarından sağlanabilir

Content -type üstbilgisi bir bileşenin veya gövdenin tamamının içeriğini tanımlamada kullanılır. Bu anahtarın değeri top-level/subtype parametreleri kullanılarak üst düzey tür mü yoksa alt tür mü olduğu bilgisini verir. Parametreler gerekli veya isteğe bağlı olabilir.

Aşağıdaki content-type örneğinde top-level in multipart olması mesajın gövdesinde birden fazla döküman olduğunu gösterir, "mixed" sub-type ise her bir dökümanın farklı tipte olduğunu gösterir.

Content-Type: multipart/mixed;

Boundary="----=_nextpart_000_00DE_01511A02.DB1A02A0"

Bu örnekte, content-type üstbilgisi elektronik posta programına bu mesajın birden fazla bileşeni olduğu ve her bileşenin karakter dizisiyle ayrıldığı söylenmektedir.

"----=_nextpart_000_00DE_01511A02.DB1A02A0"

Sınırlayıcı dize her bir bileşenin başlangıcını işaretler. Bütün durumlarda iki kısa çizgi ile başlar. Sınırlayıcı dize mesajın sonunu belirtmek için de kullanılır, . Bu durum iki kısa çizgiyle başlar ve hemen arkasından iki kısa çizgi daha gelir. Alıcı e-posta programı sınırlayıcı dizeyi iki kısa çizgi takip ettiğinde, mesajın son bileşeninin okunduğunu bilir.

Bir elektronik postanın her bir bileşeni bu sınırlayıcı dizeyle, isteğe bağlı olarak MIME bilgisi, ve zorunlu bir boş satırla başlamak zorundadır. Eğer boş satır atlanmışsa, alıcının elektronik posta programı üstbilginin nerede bittiğini ve mesajın metninin nerde başladığını söylemede zorluklarla karşılaşabilir. Metin, görüntü(resim), ses, video, uygulama, çok parçalı, ve mesaj olarak 7 çeşit top-level tipi vardır.. Başka bir content -type örneği, aşağıda verilmiştir.

Content-type: text/html; charset=euc-kr;

Content-Type: application/zip; name="testfile.zip"

İlk satır, mesajın Kore dili karakter seti kullanılarak HTML formatında olacağını belirtir. İkinci satır. Bileşenin bir zip dosyası olacağını ve testfile.zip olarak kaydedilebileceğini belirtir. İkilik sayı düzenindeki (binary) dosyalar (sıkıştırılmış arşiv gibi) eklenti olarak gönderilebilir. Böyle durumlarda, gönderen ilk olarak binary dosyayı internet üzerinde gönderebilmek için kodlamak zorundadır.

4.2 İstenmeyen Elektronik Posta (Spam)

İstenmeyen elektronik posta RFC 524 esasları üzerinde elektronik posta sistemlerinde gerçekleştirilmiş, SMTP protokolünün kötüye kullanılmasının bir şeklidir. İlk olarak 1973 de önerilen RFC 524, bilgisayar güvenliğinin belirgin bir endişe yaratmadığı bir dönemde geliştirilmiştir. RFC 524 ün güven çok güvenli bir komut seti olmaması, SMTP protokolünü kötüye kullanılmaya karşı hassas bir hale getirir.

Çoğu istenmeyen elektronik posta üretme aracı SMTP'deki güvenlik açıklarını kullanmaktadır. Bunu elektronik posta üstbilgilerinin sahtesini üreterek, gönderenin adresini ve gönderen sistemi gizleyerek yapıyorlar, öyle ki gerçek gönderenin kimliğini tespit etmek zor veya imkansızdır.

SMTP protokolünün bu açıkları kapatmak için geliştirilmiş protokollerin çoğu elektronik postayı kabul etmeden önce gönderenin kimliğini doğru olarak tespit edebilmek için özellikler içerir. Ancak bu protokollerin yaygınca kullanılması çok zordur, çünkü yeni protokolü gerçekleştirenler sadece bu protokolü gerçekleştiren diğer sistemlerden e-posta alabilir. Bu yüzden yakın gelecekte daha güvenli bir SMTP olmadan, istenmeyen elektronik posta, kuruluşları etkili bir istenmeyen elektronik posta engelleme çözümü arayıp bulmaya yönelten bir problem olmaya devam edecektir

Analistlerin tahminleri günümüzde dünyadaki e-postaların %60'ından fazlasının istenmeyen elektronik posta olduğunu gösteriyor. İstenmeyen elektronik posta artık sadece can sıkıcı bir şey değildir. İstenmeyen elektronik posta şimdi belirgin bir güvenlik sorunu ve finansal kaynaklar üzerinde ağır bir yüktür. Aslında, bu istenmeyen elektronik posta istilasının şirketlere her sene 20 milyon dolarlık bir verimlilik kaybına sebep olduğu tahmin edilmiştir.

Bugün istenmeyen elektronik posta problemini engellemeye yardımcı olmak için çok sayıda çözüm vardır. Bu çözümler elektronik postanın analizinde ve onun gerçekten istenmeyen elektronik posta olup olmadığını belirlemede farklı teknikler olarak kullanmaktadır. İstenmeyen elektronik posta sürekli değiştiğinden, en etkili istenmeyen elektronik posta engelleme çözümleri bu tekniklerden bir kaçını içermelidir.

4.3 İstenmeyen Elektronik Posta Engelleme Teknikleri

4.3.1 Kelime filtreleri

Kelime filtreleri belirgin istenmeyen elektronik postanın çoğunluğunu engellemede halen etkili basit bir yoldur. Örneğin basitçe istenmeyen elektronik postada yaygın olarak bulunan "viagra" gibi belli anahtar kelimeleri içeren e-postaları tanımlarlar. İstenmeyen elektronik postacılar sıklıkla kelime filtrelerinden kurtulmak için bilinçli olarak kelimeleri hatalı yazarlar. Bu nedenle kelime filtreleri, düzenli olarak anahtar kelimelerin varyasyonları ile düzenli olarak güncellenmelidir. Örneğin "viagra" bilinçli olarak "v1agra" olarak yazılır, bu yüzden kelime filtresi hem "viagra" hem de "v1agra"'yı içerecek şekilde güncellenmelidir.

Bazı durumlarda kelime filtreleri hatalı çıkarımlar yaratama riskiyle çalışırlar. Örneğin bir klinik araştırmacının veya eczacının içinde "viagra" kelimesi geçen normal bir elektronik postası yanlışlıkla engellenebilir.

Sonuçta, kelime filtreleri, eğer istenmeyen elektronik postacıların benzersiz yazım hataları kadar, yeni anahtar kelimeler ve ifadelerle sürekli güncellenirse etkili bir istenmeyen elektronik posta engelleme tekniği olabilir.

4.3.2 Kural tabanlı puanlama sistemleri

Kural tabanlı puanlama sistemleri kelime filtrelerinden daha komplike bir istenmeyen elektronik posta engelleme tekniğidir. (AI-Artificial Intelligence) yapay zeka sistemleri olarak da bilinen bu sistemler anahtar kelime kontrolü yapan kelime filtrelerine benzerdir. Bununla birlikte, kelime filtreleri sadece anahtar kelimeyi içeren e-postaları engellerken, kural tabanlı puanlama sistemleri elektronik postaları analiz etmek için kurallar kullanır ve buldukları her anahtar kelime için bir puan belirlerler.

Örneğin: "DISCOUNT" kelimesi içeren bir e-postada bütün büyük harfli kelimeler +2 puan alırlar. "click here" ifadesine sahip bir e-posta ise +1 puan alabilir. Ne kadar çok puan alırsa, e-postanın istenmeyen elektronik posta olma olasılığı da o kadar artar. Eğer elektronik posta belli bir puana veya eşik değerine ulaşırsa, istenmeyen elektronik posta olarak sınıflandırılır. Kural tabanlı puanlama sistemlerinde, kuralların her biri için uygun puanı belirlemede çok miktarda istenmeyen elektronik posta ve istenmeyen elektronik posta olmayan yani normal elektronik posta kullanılır.

Açık kodlu bir istenmeyen elektronik posta filtresi olan Spam Assassin, kural tabanlı puanlama sistemlerine bir örnektir. İstenmeyen elektronik postayı tespit etmek için, Spam Assassin elektronik posta üstbilgileri ve gövde metinleri üzerinde sezgisel testlerin geniş bir aralığını kullanır.

İstenmeyen elektronik postacılar ve istenmeyen elektronik posta üreten uygulamaları, durağan olmadığı için kural tabanlı puanlama sistemleri de kelime filtrelerinin karşılaştığı sorunlarla karşılaşırlar. Kural tabanlı puanlama sistemlerinin etkin kalabilmesi için kurallar düzenli olarak, sırasıyla güncellenmek zorundadır.

Örneğin, eğer kural tabanlı bir puanlama sistemi, "viagra" kelimesine puan atayan bir kurala sahipse, istenmeyen elektronik postacılar istenmeyen elektronik postalarını başarılı bir şekilde iletmek için "viagra" kelimesinde bilerek çok faklı şekillerde yazım hatası yaparak bu kuralı kolayca atlatabilirler. Kural tabanlı puanlama sistemleri, eğer doğru kullanılırlarsa, çok etkin olabilirler, % 90'nın üzerinde gelen istenmeyen elektronik postayı eleyebilirler.

4.3.3 Bayes filtreleri

Bayes filtreleri her bir kullanıcı için kişiselleştirilmiş ve istenmeyen elektronik postadaki değişikliklere otomatik olarak uyum sağlayan filtrelerdir. Bir elektronik postanın istenmeyen elektronik posta olma olasılığını belirlemek için, filtreler elektronik postadaki kelime veya ifadelerin ilgili kullanıcının daha

önceki (normal ve istenmeyen) elektronik postalarındaki kullanım sıklığı karşılaştırılmak için Bayes analizini kullanırlar.

Bayes filtreleri çok güçlüdür ve istenmeyen elektronik posta engellemede en kesin (doğru) tekniklerden birisi olarak dikkate alınırlar (Filtre iyi eğitildiğinde). Bayes filtreleri hakkındaki çoğu rapor, %99'luk bir başarının sağlandığını gösterir. Bayes filtre eğitimi için normalde ilgili kullanıcıdan aşağı yukarı 200 normal elektronik posta ve 200 istenmeyen elektronik postaya gerek duyulur. İlgili kullanıcıya ait daha önceki elektronik postaların oluşturduğu veritabanının büyüklüğü, filtrenin doğruluğunu arttırır.

4.3.4 Kara listeler

Kara liste, yaygın bir istenmeyen elektronik posta engelleme tekniğidir. Hesaplama yükü yoktur ve gerçekleştirilmesi kolaydır. Bu teknik basitçe bilinen istenmeyen elektronik postacıların IP adreslerinin bir listesini el ile tutan kurumları içerir, böylece bu adreslerden gelen elektronik postalar engellenir.

İstenmeyen elektronik postacılar düzenli olarak IP adreslerini değiştirdiğinden ve geniş aralıkta IP adresleri kullandığından, kara listeler kısa zaman aralıklarında, küçük istenmeyen elektronik posta miktarlarını engellemede çok etkindirler. Tek bir kaynağa özgü istenmeyen elektronik postayı engellemede çabuk bir çözüm sağlarlar, fakat genel bir istenmeyen elektronik posta filtreleme çözümü olarak etkin değildirler.

Kara listeye bir alternatif beyaz listedir. Beyaz liste de sadece elektronik postaların kabul edileceği adreslerin bir listesidir. Kara listenin tersi olan bu kavram, pratik değildir; çünkü kullanıcılar sadece daha önceden bildikleri adreslerden elektronik posta alabilirler, herhangi bir yeni kaynaktan elektronik posta almalarını olanaksız kılar.

4.3.5 Gerçek zamanlı kara delik listeleri

DNS RBL olarak da bilinen RBL, gelen her elektronik postanın IP adresini RBL'de bulunan ip adreslerinin bir listesiyle karşılaştırır. Eğer ip adresi RBL'nin bir parçası ise, elektronik posta istenmeyen elektronik posta olarak belirlenir ve engellenir.

Kara liste IP tekniğinden farklı olarak, RBL ler kurumlar tarafından elle güncellenmezler. RBL operatörleri herkese açık RBL lerin geliştirilmesini, bakımını yaparlar ve şirketler basitçe onlara abone olurlar.

Çoğu kuruluşlar RBL kullanmaktan memnundurlar; çünkü, DNS'e benzer bir protokol kullanarak geliştirilen bu RBL'ler hesaplama yükünü azaltır ve aynı zamanda network yükünü de azaltır.RBL'nin bir dezavantajı hatalı çıkarımlar üretmesidir. Çoğu RBL agresiftir ve rapor edilen bütün istenmeyen elektronik posta kaynaklarını bloklar. Bununla birlikte, çoğu zaman istenmeyen elektronik posta kaynağı olan Yahoo, Hotmail gibi popüler ISP ler, aynı zamanda normal, legal elektronik posta kaynağıdırlar. Bu durumlarda normal elektronik postalar, IP adresi tanımlanır tanımlanmaz reddedildiği için, genellikle alınmazlar.

RBL'ler bir kaynaktan istenmeyen veya normal elektronik posta gönderildiğinde ayırt edemezler. Sadece listelerindeki IP adreslerinden gelen her elektronik postayı engellerler, bu sebepten bazen hatalı çıkarımlar üretirler.

RBL ler istenmeyen elektronik postayı engellemede etkili olduğundan bir kurumun istenmeyen elektronik posta engelleme stratejisinin bir parçası olması önerilir. Hangi RBL'nin kullanılacağının dikkatli seçilmesi, hatalı çıkarım üretme dezavantajı olmadan etkin olarak istenmeyen elektronik posta engellenmesini sağlayabilir.

4.3.6 DNS MX kayıt araması

Bu, sahte bir from veya return adresi kullanan istenmeyen elektronik postacılardan gelen istenmeyen elektronik postaların engellenmesinde etkin bir tekniktir. İstenmeyen elektronik postacılar hileli (fake) adresler kullanırlar ve bu sayede gönderdikleri istenmeyen elektronik posta onları bulmada bir iz olamaz.

From adresinin geçerli olup olmadığını belirlemek için, sistem from adresinde kullanılan domain üzerinde bir arama yapar. Eğer domain geçerli bir MX kaydına sahip değilse, from adresi de geçerli değildir ve elektronik posta istenmeyen elektronik posta olarak etiketlenir. Benzer aramalar return adresi içinde yapılabilir.

4.3.7 Ters DNS araması

Bu, gelen elektronik postanın kaynak IP'si üzerinde yapılan ters DNS aramasını kullanan etkin bir istenmeyen elektronik posta engelleme tekniğidir. Eğer ters aramadan elde edilen domain elektronik postadaki from adresi ile eşleşirse elektronik posta kabul edilir. Eşleşmezse elektronik posta reddedilir.

Ters DNS aramaları popülerken, sıklıkla iyi çalışmazlar. Çoğu ters DNS kaydı doğru olarak herhangi bir "vanity" alan adı gibi yerleştirilmediğinden ve daha fazlası da doğru yerleştirilmiş olmadığından, çok sayıda hatalı çıkarım(False Positive) üretebilirler.

Muhtemelen doğru bir ters DNS kaydına sahip değildir. Bu tip domainlerden gelen elektronik postalara reddedilecektir ve bu da kabul edilemez derecede yüksek hatalı çıkarım oranına sebep olacaktır.

Vanity domain names; bunlar genellikle, elektronik postanın kişisel veya ailelerin kullanması için kayıt ettirilmiş alan adlarıdır. Genellikle kendi elektronik posta sunucuları olmaz, bir hosting şirketi ile bir elektronik posta sunucusunu paylaşırlar.

4.3.8 Yeni ters DNS arama sistemleri

Bir takım istenmeyen elektronik posta engelleme teknikleri DNS sistemini kullanarak, sahte gönderen adreslerinden istenmeyen elektronik posta gönderilebilmesini sınırlamak için önerilmişlerdir. Bu teknikler ters dns arama tekniği üzerine geliştirilmişlerdir. Önerilen bu teknikler :

Reverse Mail Exchanger (RMX): Bir domain sahibinin, domainleri adına ,elektronik posta göndermeye yetkilendirilmiş bütün posta sunucularını listelemesine imkan vermek için tasarlanmış bir mekanizmadır.

Sender Permitted From (SPF): Gönderenin sahtekarlığını önlemeye yardımcı olan bir SMTP uzantısıdır. Açık ve ücretsiz bir standarttır.

Designated Mailers Protocol (DMP): Bu, kendi domaininiz adına elektronik posta göndermede MTA veya posta sunucularını yetkilendirmek için önerilmiş bir standarttır. Bu domaininizin istenmeyen elektronik postacılar veya virüsler tarafından kötüye kullanılmasını engeller. Yahoo! Domain Keys , Microsoft Caller ID for Email

Bu yaklaşımlar çoğu bakımdan benzerdir. DNS MX kayıt aramasına benzer olarak, bu ters arama çözümleri belli bir domain den gelen elektronik postanın o domain için izin verilmiş bir IP den mi geldiğini belirlemek için ters MX kayıtları tanımlarlar. Doğru RMX/SPF/DMP adres aralıklarından kaynaklanmayan (gelmeyen) elektronik posta adresleri hileli ve elektronik posta da istenmeyen elektronik posta olarak belirlenir.

Ters DNS arama gibi bu tekniğin de vanity alan adları ile problemleri vardır. Fakat kısmen düzeltilmişlerdir. Genel durum, ISP'lerinki yerine kendi alan adlarını kullanmak isteyen, fakat kendi IP adreslerini veya sunucularını almaya gücü yetmeyen küçük şirketleri veya bireysel kullanıcıları içerir .

Bunlar kayıtlı domain adlarından elektronik posta göndermek için uygulamalarını basitçe yapılandırırlar. Maalesef gönderenin IP adresinin bir araması gönderenin domainini bulmaz, ve gönderenin domainin bir araması doğru bir ters MX kaydı bulamayabilir. Mobil, çevirmeli ağ ve IP adresini sıkça değiştiren diğer kullanıcılar için durum böyledir.

4.3.9 Honeypots

Honeypotlar veya tuzak elektronik posta sistemleri, çok miktarda istenmeyen elektronik posta toplamak için kullanılır. Bu tuzak elektronik posta adresleri gerçek bir son kullanıcıya ait değildir, adresin legal olduğunu düşünen istenmeyen elektronik postacıları çekmek için genel olarak yapılmışlardır. İstenmeyen elektronik posta toplandığında, hashing sistemleri veya fingerprinting gibi belirleme teknikleri istenmeyen elektronik posta işleme ve bilinen istenmeyen elektronik postanın bir veritabanını yaratmada kullanılır.

4.3.9.1 Hashing sistemleri

Hashing sistemleriyle, her bir istenmeyen elektronik posta, istenmeyen elektronik postanın içeriğine uyan (tekabül eden) bir belirleme numarası veya "hash" alır . Sonra bilinen istenmeyen elektronik posta e-postaların bir listesi ve tekabül eden hash yaratılır. Bütün gelen e-postalar bilinen istenmeyen elektronik postaın bu listesi ile karşılaştırılır. Eğer hashing sistemi gelen bir elektronik postanın istenmeyen elektronik posta listesindeki bir e-posta ile eşleştiğini belirlerse elektronik posta geri çevrilir(kabul edilmez). Bu teknik istenmeyen elektronik postacılar aynı veya hemen hemen aynı elektronik postaları defalarca gönderdiği sürece çalışır. Bu tekniğin orijinal uygulamalarından biri Razor olarak bilinir.

4.3.9.2 Fingerprinting

Parmak izi tekniği daha önceden istenmeyen elektronik posta olarak belirlenmiş bir e-postanın karakteristiğini veya parmak izini inceler ve bu bilgiyi aynı veya benzer e-postayı belirlemede kullanır. Bu gerçek zamanlı parmak izi kontrolleri sürekli olarak güncellenir ve hemen hemen hatasız bir sonuçla istenmeyen elektronik postayı belirlemenin bir yolunu gösterir. Parmak izi teknikleri mesaj içinde yer alan URL'lere özellikle bakar ve daha önceden

istenmeyen elektronik posta propagator olarak belirlenmiş URL'lerle karşılaştırır. Hashing'le honeypot'lar veya fingerprinting büyük ölçüde benzer istenmeyen elektronik postalar gönderildiğinde etkindir. Eğer her bir istenmeyen elektronik posta benzersiz olarak yapılırsa, bu teknik zorluklarla karşılaşacak ve başarısız olacaktır.

4.3.10 Challenge/Response sistemleri

Challenge/response sistemleri, her gün milyonlarca istenmeyen elektronik posta üretmek için otomatik elektronik posta programları kullanan istenmeyen elektronik postacılara karşı koymak için kullanılırlar. Bu sistemler gelen istenmeyen elektronik posta için barikatlar kurarak istenmeyen elektronik postacıları yavaşlatmaya yönelik tasarlanmışlardır.

Challenge /Response sistemleri, istenmeyen elektronik posta Arrest veya Mailblocks tarafından önerilen, izin verilen gönderenleri bir listesini oluştururlar. Yeni bir gönderenden her elektronik posta geldiğinden bu challenge/response sistem kullanıcısına gider, elektronik posta teslim edilmeden önce geçici olarak tutulur. Challenge/response sistemi elektronik posta gönderene bir challange yollar. Bu challange genellikle bir Url'ye olan bir link veya yanıt elektronik postasındaki bir kutunun içine gönderenin kopyalayacağı bir sayısal kod isteğinden oluşur. Eğer gönderen "challenge" yi başarılı olarak tamamlarsa, challenge/response sistemi onu izin verilmiş kullanıcılar listesine ekler ve bu e-posta ilgili hedefe teslim edilir.

Challenge response sistemleri, istenmeyen elektronik postacıların challenge'yi almayacak hatalı elektronik posta adresleri kullandıkları, ve gerçek elektronik posta adresi kullanan istenmeyen elektronik postacıların bütün challenge'leri yanıtlamayacağı varsayımları altında çalışır.

4.3.11 Antivirüs taraması

Anti-virüs taraması kendi kendine çoğalmaya çalışan virüs programları tarafından üretilen istenmeyen elektronik postaların sayısının fazlalığından dolayı istenmeyen elektronik posta engellemenin bir metodu olarak görülebilir. Bir virüs tarama çözümü bir organizasyonun genel istenmeyen elektronik posta filtreleme çözümünün bir parçasını oluşturmada etkili bir araçtır.

İstenmeyen elektronik posta, şirketler için milyonlarca dolar verimlilik kaybına mal olan, günden güne büyüyen bir problemdir. Bununla birlikte istenmeyen elektronik postayı engellemede kullanılan çeşitli istenmeyen elektronik posta engelleme teknikleri vardır.

İstenmeyen elektronik postacılar istenmeyen elektronik posta göndermede kullandıkları metotları sürekli değiştirerek istenmeyen elektronik posta filtreleme tekniklerini baypas etmeye çalışırlar. Şirketler için en iyi olanı birden fazla istenmeyen elektronik posta engelleme tekniği kullanmaktır. Bu tekniklerden her birinin avantajları, dezavantajları ve sınırlamaları vardır. Bir organizasyona giren istenmeyen elektronik postaları en aza indirmek için, istenmeyen elektronik postayı engellemede etkili tekniklerin kombinasyonu uygulanmalıdır.

5. BAYES YAKLAŞIMI VE UYGULAMASI

5.1 Belge Sınıflandırma (Metin Sınıflandırma)

Belge sınıflandırma (metin sınıflandırma veya kategorize etme olarak da bilinir), içeriğine dayanarak bir belgeyi bir veya daha fazla kategoriye ayırma işlemidir. Gelen metin belgeleri, bir talim kümesinde etiketlenmiş belgelerden çıkarılan bilgiye dayanarak oluşturulmuş, önceden tanımlanmış kategorilere atanır. Otomatik belge sınıflandırma belgenin karakteristiklerine uygun olarak oluşturulmuş bir sınıflandırıcı vasıtasıyla gerçekleştirilen tümevarımlı yönlendirmeli öğrenme işlemidir. Otomatik belge sınıflandırma, Bilgi Geri Kazanımı (Information Retrieval) ve Yapay Zeka'nın (Artifical Intelligence) ortak bir alanı olan Makine Öğrenmesi (Machine Learning) tekniklerini beraberce kullanır.

Otomatik Öğrenme, verinin analiziyle uğraştığı için ağırlıklı olarak istatistikle örtüşür ve arama motorları, tıbbi teşhisler, stok pazar analizleri, DNA sıralarını sınıflandırma, el yazısı tanıma, oyun oynama ve robot hareketleri v.s gibi geniş bir alanda kullanılır. Otomatik öğrenme yaygın olarak yönlendirmeli ve yönlendirmesiz öğrenme olmak üzere ikiye ayrılır.

5.2 Yönlendirmeli Öğrenme

Yönlendirmeli öğrenmede algoritma, talim verisinden öğrenicinin sınıflandırma işleminde kullanacağı bir fonksiyon üretir. Talim verisi girdi nesneleri ve istenen çıktıların çiftlerinden oluşur. Fonksiyonun çıktısının girdi nesnesinin sınıf etiketini öngörmesiyle sınıflandırma yapılır. Yönlendirmeli öğrenicinin görevi az sayıda talim verisini yani girdi ve hedef çıktı çiftlerini gördükten sonra görülmemiş bir durum için fonksiyonun değerini tahmin etmektir.

Bir yönlendirmeli öğrenme işlemi çeşitli adımlardan oluşmaktadır. Öncelikle derlem örneklerinin tipi belirlenir. Derlem tipi belirlendikten sonra, yönlendirmeli öğrenmeyi gerçekleştirecek sınıflandırıcının öğrenmesi için talim kümesini ve performansının değerlendirilmesi için test kümesini oluşturacak örnekler

seçilir. Bir sonraki adımda fonksiyonun girdi özelliğinin ne şekilde gösterileceği yani belgenin nasıl temsil edileceği belirlenir.

Belgeler tipik olarak kelimelerden oluşur. Belgeler, içinde bulunan kelimeler ve onların oluş frekanslarının yer aldığı bir öznitelik vektörü ile temsil edilir. Gereksiz ve büyük öznitelik vektörlerinden sakınmak için belgelere genellikle düşük bilgili kelimeleri elemek için "işlev- kelimeler"(stop-words) ve "kelime gövdeleri" (word stems) teknikleri uygulanır. Bu işlemlerden sonra vektör uzaylarının boyutunu küçültmede özellik seçimi uygulanır. Özellik seçiminde yaygın olarak kullanılan metotlardan bazıları; terim ve kategoriler arasındaki bağımlılığı ölçen "ki-kare" dağılımı(Chi Square), kelimelerin birlikteliklerinin ve ilgili uygulamaların bir istatistiksel dil modellemesi olan "ortak bilgi" (Mutual Information), bir terimin en iyi olduğunu tanımlayan "bilgi kazanım" kriteridir.(Information Gain) .

Belgeler öznitelik vektörleri olarak temsil edildikten sonra belgeleri sınıflandırmak için probleme uygun öğrenme algoritmaları seçilir. Belge sınıflandırmada kullanılan Decision Tree, Naive Bayes, Support Vector Machine, Rocchio v.b çok sayıda öğrenme algoritması vardır. Algoritma belirlendikten sonra sınıflandırıcının performansı test kümesi üzerinde ölçülür.Belge sınıflandırmanın etkinliği ölçülürken sıklıkla "duyarlık ve anma", "doğruluk ve hata" analizleri kullanılır.

"Kaynak Özetleri" kısmında da belirtildiği gibi belge sınıflandırma veya metin sınıflandırma kavramı altında farklı otomatik öğrenme algoritmaları kullanarak problemi yönlendirmeli öğrenme yaklaşımı olarak ele alan çeşitli yayınlar vardır. Bu yayınlarda Otomatik öğrenme algoritmalarından, uygulaması basit ve çoğu durumda etkili sonuçlar veren "Naive Bayes" sıkça kullanılmış ve Yalın Bayes Sınıflandırmaya "Naive Bayes Classification" ek olarak Bayes İstenmeyen E-posta Filtreleme (Bayesian Spam Filtering) kavramının gelişmesinde etkili olmuştur. Bayesian Spam Filtering veya Bayes Filtering kavramlarının Naive Bayes Classification kavramı altında ele alınması tartışma konusu olabilir.Bu çalışmada istenmeyen elektronik postaların filtrelenmesi problemi, Bayes İstenmeyen Elektronik Posta Filtreleme kavramına dayandırılmıştır.

5.3 Yalın Bayes Sınıflandırma (Naive Bayes)

Bir olasılıksal sınıflandırma olan Yalın Bayes sınıflandırmanın ana fikri, bir belgenin sınıfının olasılığını tahmin etmek için verilen bir kelimenin sınıfının koşullu olasılıklarını kullanmaktır. Belge sınıflandırma gibi bazı öğrenme problemlerinde yaygın olarak kullanılan en pratik yaklaşımdır. Bu yaklaşımın "Yalın" (Naive) kısmı içindeki kelime bağımsızlığı varsayımından kaynaklanmaktadır. Çünkü kelime kombinasyonlarının olasılıklarını tahminci olarak kullanmaz. Bu yüzden Karar Ağacı (Decision Tree) gibi algoritmaların üstel karmaşıklığından öte verimli bir yaklaşımdır ve performansı Yapay Sinir Ağları(Neural Networks) ve Karar Ağacı ile karşılaştırılabilir.

Yalın Bayes'de Artımlı (Incremantal) olarak tabir edilen online bir öğrenme durumu vardır; her bir talim örneği artımlı olarak bir hipotezin doğru olma olasılığını arttırır veya azaltır. Öncül bilgi gözlemlenen verilerle birleştirilebilir.

Bayes kuralı;

A ve B rasgele olaylar olsun;

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

$P(A)$: A olayının bağımsız olasılığı prior (öncül) olasılık

$P(B)$: B olayının bağımsız olasılığı

$P(B \mid A)$: A olayının olduğu bilindiğinde B olayının olasılığı likelihood (şartlı olasılık)

$P(A \mid B)$: B olayının olduğu bilindiğinde A olayının olasılığı posterior (artçıl) olasılık

Bayes kuralına dayanarak $P(A \mid B)$ yi maksimum yapan durumlar hesaplanabilir.

"E" A olayının bütün durumlarının kümesi;

$$A_{MAX} = \arg\max_{A \in E} P(A \mid B)$$

$$= \arg\max_{A \in E} \frac{P(B \mid A)P(A)}{P(B)}$$

$$= \arg\max_{A \in E} P(B \mid A)P(A)$$

Burada $P(B)$ yi sabit olarak göz ardı edilebilir. Sınıflandırıcının görevi yeni bir örnek için doğru sınıfı tahmin etmek olacaktır.

Sınıflandırma problemi öncül (a-priori) olasılıklar kullanılarak şu şekilde formüle edilebilir:

$P(v \mid X)$ = örnek durum için olasılık

$X=<a1,\ldots,ak>$ v sınıfının örnekleri

P(sınıf=istemeyen elektronik posta | kelime1=bedava, kelime2=mp3,…)

X örneğine $P(v \mid X)$ olasılığını maksimum yapan sınıf etiketi atanır.

Bayes Teoremi:

$P(v \mid X) = P(X \mid v)\, P(v) / P(X)$

$P(X)$ bütün sınıflar için sabittir

$P(v)$ = sınıfın bütün örneklere karşı göreli frekansı

$P(v \mid X)$maksimum= $P(X \mid v)P(v)$maksimum

Problem: $P(X|v)=P(a1,\ldots,ak \mid v)$ yı hesaplamak çok zordur, çok fazla ihtimal gerektirir. (2*2k!)

Kelime bağımsızlığı varsayımı (yalınlık)

$P(a1,\ldots,ak \mid v) = P(a1 \mid v)\cdot\ldots\cdot P(ak \mid v)$

Eğer i. özellik kesikli değerde ise $P(ai \mid v)$, v sınıfında i. özellik olarak ai değerine sahip örneklerin göreli frekansı olarak tahmin edilir.

Eğer i.ci özellik sürekli ise;

1) kesikli hale getirilebilir

2) veya $P(ai|v)$ Gauss (Normal) dağılımına sahip olduğu varsayılarak tahmin edilir, sadece ortalama ve varyans gereklidir.

İki durumda da hesaplanabilir.

Bu durum, klasikleşmiş bir örnek olan, bir tenis maçının oynanma olasılığı örneği ile ele alınmıştır. Verilen hava şartlarına göre bir tenis maçının oynanıp oynanmayacağı belirlenmektedir. Tablo 2.1, 2.2 ve 2.3'de örnek veri kümesi, frekans ve olasılıklar verilmiştir.

TABLO 2.1 Örnek veri kümesi

GÖRÜNÜM	SICAKLIK	NEM	RÜZGAR	OYUN
Güneşli	sıcak	yüksek	Yok	hayır
Güneşli	sıcak	yüksek	Var	hayır
Bulutlu	sıcak	yüksek	Yok	evet
Yağmurlu	ılık	yüksek	Yok	evet
Yağmurlu	serin	normal	Yok	evet
Yağmurlu	serin	normal	Var	hayır
Bulutlu	serin	normal	Var	evet
Güneşli	ılık	yüksek	Yok	hayır
Güneşli	serin	normal	Yok	evet
Yağmurlu	ılık	normal	Yok	evet
Güneşli	ılık	normal	Var	evet
Bulutlu	ılık	yüksek	Var	evet
Bulutlu	sıcak	normal	Yok	evet
Yağmurlu	ılık	yüksek	Var	hayır

Tablo 2.2 Frekanslar

GÖRÜNÜM	Oyun evet	Oyun hayır	SICAKLIK	Oyun evet	Oyun hayır	NEM	Oyun evet	Oyun hayır	RÜZGAR	Oyun evet	Oyun hayır	OYUN	
												evet	hayır
Güneşli	2	3	Sıcak	2	2	Yüksek	3	4	Yok	6	2	9	5
Bulutlu	4	0	Ilık	4	2	Normal	6	1	Var	3	3		
Yağmurlu	3	2	Serin	3	1								

Tablo 2.3 Olasılıklar

GÖRÜNÜM	Oyun evet	Oyun hayır	SICAKLIK	Oyun evet	Oyun hayır	NEM	Oyun evet	Oyun hayır	RÜZGAR	Oyun evet	Oyun hayır	OYUN	
												evet	hayır
Güneşli	2/9	3/5	Sıcak	2/9	2/5	Yüksek	3/9	4/5	Yok	6/9	2/5	9/14	5/14
Bulutlu	4/9	0/5	Ilık	4/9	2/5	Normal	6/9	1/5	Var	3/9	3/5		
Yağmurlu	3/9	2/5	Serin	3/9	1/5								

Bu örnek veriler eşliğinde aşağıdaki yeni durum sınıflandırılırsa, Şekil 1.1' deki duruma ulaşılır.

Görünüm Sıcaklık Nem Rüzgar Oyun

Güneşli Serin Yüksek Var ?

$$v_{YB} = \underset{v_j \in \{evet, hayır\}}{\arg\max}\ P(v_j) \prod_i P(a_i \mid v_j)$$

$$= \underset{v_j \in \{evet, hayır\}}{\arg\max}\ P(v_j)\, P(\text{görünüm} = \text{güneşli} \mid v_j) P(\text{sıcaklık} = \text{serin} \mid v_j) .....$$

Öncül Olasılıklar (2)

Şartlı Olasılıklar (8)

Şekil 1.1 Yalın Bayes sınıflandırma formülü

Anafikir: talim kümesindeki olasılık dağılımına dayanarak her bir sınıf için olasılık hesaplanır. Önce her bir özelliğin olasılığı hesaba katılır, bütün özellikler eşdeğer önemde ele alınır ve olasılıklar çarpılır.

P(EVET) = 2/9 3/9 3/9 3/9 = 0.0082

P(HAYIR) = 3/5 1/5 4/5 3/5 = 0.0577

Verilen bir sınıfın toplam olasılığını hesaba katılır ve özelliklerin olasılıklarıyla çarpılır ;

P(EVET) = 0.0082 . 9/14 =0.0053

P(HAYIR) =0.0577. 5/14 =0.0206

Bu olasılığı maksimum yapan sınıf seçilirse, yeni durum "HAYIR" olarak sınıflandırma anlamına gelecektir.

Bu olasılıkları tek bir olasılığa normalleştirmek gerekirse,

$$\frac{0.0206}{0.0206+0.0053}=0.795$$

elde edilir. Küçük talim kümeleri veya sadece bir sınıfta yer alan özelliklerin olasılıklarını hesaplarken problemlerle karşılaşmamak için m-estimate formülü kullanılır.

$$P(a_i \mid v_j)=\frac{n_C+mp}{n+m}$$

n_c: a_i özelliği ve v_j sınıfındaki örneklerin sayısı

p: olasılığın öncül tahmini

m:eşdeğer talim kümesi (sabit)

diğer bilgilerin yokluğunda , tekbiçimli bir öncül varsayılır

$p=\frac{1}{k}$ burada k a_i özelliğinin alabildiği değerlerin sayısıdır.

Yalın Bayes sınıflandırma metin veya elektronik posta problemlerine uygulanırken;

Örneğin içeriği "selam merhaba nasılsın" olan bir elektronik posta için formül aşağıdaki gibidir.

k_1,k_2,k_3 özellikler yani kelimelerimiz olsun

$s_j \in (\text{istenmeyen}, \text{normal})$ S sınıf kümemiz olsun.

$$S_{NB} = \arg_{s_j \in (\text{istenmeyen},\text{normal})} \max P(s_j) \prod_i P(k_i \mid s_j)$$

$$= \arg_{s_j \in (\text{istenmeyen},\text{normal})} \max P(s_j) P(k_1 = \text{selam} \mid s_j) P(k_2 = \text{merhaba} \mid s_j) P(k_3 = \text{nasılsın} \mid s_j)$$

Talim aşamasında:

$P(s_j) = \frac{n}{N}$ prior olasılıklar hesaplanır.

Likelihood olasılıklar şu şekilde hesaplanır.

$$P(k_i \mid s_j) = \frac{n_i + 1}{n + \mid KH \mid}$$

N: bütün elektronik postalardaki kelime sayısı

n: s_j sınıfındaki kelime sayısı

n_i : k_i kelimesinin sınıfında oluş sayısı

$\mid KH \mid$: kelime haznesinin büyüklüğü

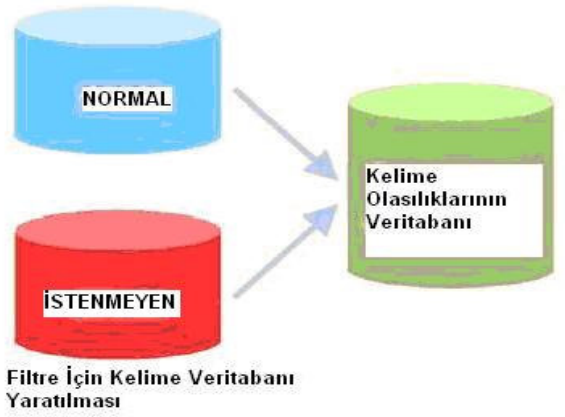
Her sınıf için olasılıklar hesaplanır ve örnek olasılığı yüksek olan sınıfa atanır. Yalın Bayes sınıflandırmada kelimelerin birbirinden bağımsız olduğu varsayılır. Aslında durum böyle değildir fakat kelimelerin bağımlı olduğu bir modeli hesaplamak son derece zordur genellikle talim kümesinde (veritabanında) her bir özellik kombinasyonu için yeteri kadar örnek yoktur.

5.4 Bayes İstenmeyen Elektronik Posta Filtreleme (Bayesian Spam Filtering)

Bayes istenmeyen elektronik posta filtreleme, Bayes istatistiksel metotlarının elektronik postaları normal veya istenmeyen olarak sınıflandırmak için kullanılması işlemidir. 1998 yılında Sahami (Sahami, 1998) tarafından önerilmiş ve 2002 yılında yayınlanan Paul Graham' in "A Plan for Spam" makalesi ile tanınmıştır. Paul Graham, Bayes teoremini istenmeyen elektronik posta ayıklama problemine direk olarak uygulamış ve elde ettiği sonuçlar son derece başarılı olmuştur. Bugün çoğu modern elektronik posta istemci ve sunucu tabanlı elektronik posta filtresi Bayes İstenmeyen Elektronik Posta Filtreleme uygulamaktadır.

Bayes filtreleme, "çoğu olay bağımlıdır ve gelecekte olacak bir olayın olasılığı o olayın daha önceki oluşlarından çıkarılabilir" prensibine dayanmaktadır. Bu prensip istenmeyen elektronik postanın sınıflandırılmasında kullanılabilir. Eğer

bir metnin bir kısmı normal elektronik posta içinde değil de, sıklıkla istenmeyen elektronik posta içinde yer alıyorsa, bu durum bu elektronik postanın istenmeyen elektronik posta olduğunun varsayımı kabul edilebilir. Elektronik posta bu yöntem kullanılarak filtrelenmeden önce, şekil 2.1 de gösterildiği gibi kullanıcının istenmeyen ve normal elektronik postalarından toplanmış kelimelerle bir veritabanı oluşturulması lazımdır.



Şekil 2.1 Filtre için kelime veritabanı yaratılması

Olasılık değeri sonra her bir kelimeye atanır; olasılık o kelimenin istenmeyen elektronik posta içinde normal elektronik postaya zıt olarak hangi sıklıkla geçtiğini değerlendiren hesaplar üzerine temellendirilmiştir.

Bu, kullanıcının giden elektronik posta analizi ve bilinen istenmeyen elektronik posta analizi ile yapılır: iki elektronik posta havuzu içindeki bütün kelimeler, elektronik postanın istenmeyen elektronik posta olduğunu gösteren belirli bir kelime olasılığı üretmek için analiz edilir.

Normal elektronik posta analizinin bir kurumun veya kişinin elektronik postaları üzerinde gerçekleştirilmesi ve kuruma veya kişiye uyarlanması önemlidir. Örneğin; bir finansal kurum “banka” kelimesini birçok kez kullanabilir ve eğer genel bir istenmeyen elektronik posta kural kümesi kullanılıyorsa birçok hatalı çıkarım

elde edilebilir. Diğer taraftan Bayes filtresi, eğer başlangıç periyodundan geçerek, kuruma göre uyarlanmışsa kurumun normal (geçerli) giden maillerini dikkate alır("banka"yı normal mesajlarda sıklıkla kullanılan bir kelime olarak tanır), ve böylece daha iyi bir istenmeyen elektronik posta belirleme oranı ve düşük hatalı çıkarım oranı sağlar.

Normal elektronik postaya ilave olarak, Bayes filtresi istenmeyen elektronik posta veri dosyasına da ihtiyaç duyar. Bu istenmeyen elektronik posta veri dosyası çok sayıda bilinen istenmeyen elektronik posta örneği içermeli ve istenmeyen elektronik posta filtreleme yazılımı tarafından sürekli yeni istenmeyen elektronik postalarla güncellenmelidir. Bu, Bayes filtresinin en yeni istenmeyen elektronik posta hilelerinin farkında olmasını sağlar ve sonuç olarak yüksek istenmeyen elektronik posta engelleme oranı sağlar.

Gerçek filtreleme şöyle yapılır; normal ve istenmeyen elektronik posta veritabanları yaratıldığında, kelime olasılıkları hesaplanabilir ve filtre kulanım için hazır duruma gelir. Yeni bir elektronik posta geldiğinde, bu elektronik posta kelimelere ayrılır. Elektronik postanın istenmeyen olup olmadığını tanımlamada anlamlı olan, en çok alakalı kelimeler seçilir. Bu kelimelerden, Bayes filtresi yeni gelen bu mesajın istenmeyen elektronik posta olup olmadığının olasılığını hesaplar. Eğer olasılık bir eşik değerinden büyükse, örneğin 0,9, mesaj istenmeyen olarak sınıflandırılır.

İstenmeyen elektronik posta için Bayes yaklaşımı çok verimlidir ve çok düşük hatalı çıkarım oranı ile birlikte %99,7 oranında istenmeyen elektronik posta belirleme başarısı yakalamıştır (BBC, 2003).

Kelime olasılığı şu şekilde hesaplanır: Örneğin ; "merhaba" kelimesi 1000 adet istenmeyen elektronik postanın 400'ü içinde geçerse ve 500 normal elektronik postada 5 tanesi içinde geçerse onun istenmeyen elektronik posta olasılığı aşağıda hesaplandığı üzere 0,9756 'dur.

S (k) = k kelimesinin istenmeyen elektronik postalarda görülme olasılığı

N (k) = k kelimesinin normal elektronik postalarda görülme olasılığı

P (k) = k kelimesi içeren bir elektronik postanın istenmeyen elektronik posta olma olasılığı

$$S(k) = \frac{\text{k kelimesini içeren istenmeyen elektronik posta sayısı}}{\text{toplam istenmeyen elektronik posta sayısı}}$$

$$N(k) = \frac{\text{k kelimesini içeren normal elektronik posta sayısı}}{\text{toplam normal elektronik posta sayısı}}$$

$$P(k) = \frac{S(k)}{S(k) + N(k)}$$

$$P(k) = \frac{0,4}{0,4 + 0,01} = 0,9756$$

Yukarıdaki P(k) şartlı olasılıkları veritabanında bulunan her bir kelime için hesaplanır. Daha sonra yeni gelen bir elektronik posta kelimelere ayrılarak bu kelimelere karşılık gelen olasılık değeri P(k) atanır.

Değerlendirilecek elektronik postanın istenmeyen elektronik posta olma olasılığı bu P(k) olasılıklarının Bayes teoremi yoluyla birleştirilmesi sonucunda hesaplanır ve bu hesaplanan değer belli bir eşik değerinin üzerinde ise elektronik posta mesajı istenmeyen elektronik posta olarak sınıflandırılır. Bayes teoremi birden fazla olasılığın birleştirilmesini mümkün kılmaktadır.

a,b,…n çeşitli olayların olasılıkları olsun. Bu olayların birlikte gerçekleşme olasılığı "p" şu şekilde hesaplanır:

$$p = \frac{a.b.c.....n}{a.b.c.....n + (1-a)(1-b)(1-c).....(1-n)}$$

Örneğin "e" bir istenmeyen elektronik posta, "i" istenmeyen elektronik posta olsun k_1 , k_2, k_3 de içerdiği kelimeler olsun bu durumda Bayes formülünü kullanarak bu elektronik postanın istenmeyen elektronik posta olma olasılığı aşağıdaki şekilde hesaplanır.

$$P(i \mid e_{(k1,k2,k3)}) = \frac{p(k_1 \mid i)p(k_2 \mid i)p(k_3 \mid i)}{p(k_1 \mid i)p(k_2 \mid i)p(k_3 \mid i) + (1 - p(k_1 \mid i))(1 - p(k_2 \mid i))(1 - p(k_3 \mid i))}$$

Ve P(i|e) olasılığı belli bir eşik değerinden büyükse ilgili elektronik posta spam olarak sınıflandırılır.

5.5 Uygulama (Prototip Filtre)

Uygulama aşamasında Bayes filtreleme yönteminin, farklı yollarla Türkçe elektronik posta mesajlarına uygulanmasını gerçekleştirmek üzere Borland Delphi 7.0 programlama dili aracılığıyla bir Bayes filtre prototipi geliştirilmiştir. Bu uygulama gerçekleştirilirken yapılan işlemler sırasıyla açıklanmaktadır. Geliştirilen prototipin kaynak kodu Ek 1' de verilmiştir.

5.5.1 Derlem oluşturulması

Öncelikle problemimiz istenmeyen elektronik postaların filtrelenmesi olduğu için, otomatik metin sınıflandırma kısmı filtremizin talim ve öğrenme işlemini gerçekleştireceği daha sonra da filtreleme sonuçlarının başarısının ölçüleceği bir elektronik posta derlemine ihtiyaç vardır. Çalışmamızda kullanılan derlem, çeşitli kullanıcıların elektronik posta hesaplarından derlenen 767 adet istenmeyen ve 1620 adet de normal elektronik posta olmak üzere toplam 2387 adet elektronik posta mesajı içeren, çoğunluğu metin içerikli ve birbirine eş olmayan mesajlardan oluşan bir derlemdir. Mesajların tamamı Türkçe içeriklidir.

Kullanılan derlemin büyüklüğü ve mesaj tarihlerinin geniş bir zamanı kapsaması başarım açısından oldukça önemli kriterlerdir. Derlemin büyüklüğü, içerdiği mesajların hangi tarihler arasında gönderildiği filtrenin görebileceği örnek sayısını etkileyeceği için öğrenme üzerinde direkt etkili olacaktır.

Örneğin herhangi bir kullanıcının 1 aylık bir mesaj trafiği o kullanıcının genel elektronik posta mesajlaşma karakteristiğini iyi bir şekilde yansıtmayacaktır. 6 aylık veya 1 yıllık bir süre öğrenme açısından daha farklı durumları kapsayacağı için daha makul olabilir.

5.5.2 Kelimelere ayırma

Derlemde bulunan mesajlar gövde metni, “konu” ve “kimden” üstbilgileri de dahil olmak üzere ele alınmış ve kelimelere ayrılmıştır. Kelimelere ayırma işlemi gerçekleştirilirken 3 taneden az harf içeren kelimeler göz ardı edilmiş ve büyük harfler küçük harfe çevrilmiştir. Kelime olarak içerisinde noktalama işareti veya rakam içeren ifadelere yer verilmemiştir. Elektronik posta mesajlarında veya internet üstbilgilerinde yer alan elektronik posta adresleri de ayıklanarak kelimelere dönüştürülmüş böylece statik istenmeyen elektronik posta filtrelemede olduğu gibi belli bir gönderenden gelen veya belli bir konusu olan mesajların engellenmesine benzer bir etki otomatik olarak oluşturulmuştur. Bununla birlikte html içerikli mesajlar için herhangi bir html etiket ayıklaması yapılmamıştır.

Html içerisinde yer alan çeşitli kodlamalar veya metinler bir gösterge olarak alınmıştır. Bunlar kelime olarak anlamlı olmasalar bile neticede bütün mesajlarda filtre bunları benzer şekilde algılayacağı için kelimeler gibi bir gösterge olarak kullanılabileceklerdir. Ayıklanan kelimeler üzerinde herhangi bir morfolojik çalışma yapılmamıştır, anlamlı olup olmadıkları incelenmemiştir, neticede hepsi birer gösterge niteliği taşımaktadır ve bir sonraki oluşlarında benzer olma eğilimindedirler.

5.5.3 Kelime olasılıklarının hesaplanması

Kelime olasılıkları hesaplanırken temel olarak iki ayrı model üzerinde çalışılmıştır. İlk olarak daha öncede belirtildiği gibi belli bir kelimeyi içeren elektronik postaların bir sınıfa ait toplam elektronik posta sayısına oranları belirlenmiştir.

S (k) = k kelimesinin istenmeyen elektronik postalarda görülme olasılığı

N (k) = k kelimesinin normal elektronik postalarda görülme olasılığı

P (k) = k kelimesi içeren bir elektronik postanın istenmeyen elektronik posta olma olasılığı

$$S(k) = \frac{\text{kelimesini içeren istenmeyen elektronik posta sayısı}}{\text{toplam istenmeyen elektronik posta sayısı}}$$

$$N(k) = \frac{\text{k kelimesini içeren normal elektronik posta sayısı}}{\text{toplam normal elektronik posta sayısı}}$$

$$P(k) = \frac{S(k)}{S(k) + N(k)}$$

İkinci model olarak kelimelerin elektronik postalarda geçme sayısı da göz önüne alınmış, istenmeyen elektronik posta olma olasılığı ise gene aynı yolla hesaplanmıştır.

$$S(k) = \frac{\text{kelimenin istenmeyen elektronik postalarda geçme sayısı}}{\text{toplam istenmeyen elektronik posta sayısı}}$$

$$N(k) = \frac{\text{kelimenin normal elektronik postalarda geçme sayısı}}{\text{toplam normal elektronik posta sayısı}}$$

$$P(k) = \frac{S(k)}{S(k) + N(k)}$$

Bu hesaplamalar kelimeler için 0.0 ile 1.0 arasında değerler elde etmemizi sağlayacak ve 0.5 değeri kelimenin nötr olduğunu 0.5'den büyük değerler 1.0'a kadar kelimenin istenmeyen elektronik postaya yatkınlığını, 0.5'den 0.0'a kadar uzanan değerler ise kelimenin normal elektronik postaya yatkınlığını ifade edecektir.

Yalnızca istenmeyen elektronik postalarda geçen ve normal elektronik postalarda geçmeyen veya normal elektronik postalarda geçen istenmeyen elektronik postalarda geçmeyen kelimeler için farklı bir durum söz konusudur. Bunlar için p(k) değeri "0" veya "1" mutlak olasılıklar olarak hesaplanacaktır ki bu istenilen bir durum değildir. Örneğin istenmeyen elektronik postalarda geçen fakat normal elektronik postalarda hiç geçmemiş bir kelimeyi ele alacak olursak bu kelimenin

ilerde normal bir elektronik postada geçmeyeceği hiçbir zaman %100 garanti değildir. Buradaki problemi gidermek için Yalın Bayes sınıflandırmadaki m-estimate'e benzer bir formül kullanılarak kelime olasılıkları şu şekilde hesaplanmıştır:

İlk modelimizde;

N: kelimenin geçtiği belli bir sınıfa ait elektronik posta sayısı olarak; kelime, normal elektronik posta mesajlarında geçiyor, istenmeyen elektronik posta mesajlarında geçmiyorsa

$$p(s)=\frac{0.5}{1+N}$$

Eğer kelime istenmeyen elektronik posta mesajlarında geçiyor normal elektronik posta mesajlarında geçmiyorsa

$$p(s)=1-\frac{0.5}{1+N}$$

N: kelimenin belli bir elektronik posta sınıfındaki mesajlarda geçme sayısı olmak üzere kelime normal elektronik posta mesajlarında geçiyor, istenmeyen elektronik posta mesajlarında geçmiyorsa

$$p(s)=\frac{0.5}{1+N}$$

Eğer kelime istenmeyen elektronik posta mesajlarında geçiyor normal elektronik posta mesajlarında geçmiyorsa

$$p(s)=1-\frac{0.5}{1+N}$$

Filtreleme işlemi sırasında filtre ilk defa gördüğü bir kelimeye nötr değer olan 0,5'i atayacaktır. Zipf 'in "Law-based" analizlerine (Zipf, 1949) benzer şekilde toplamda 5 defadan az geçen kelimeler dikkate alınmamıştır. Bu kelimelerin mesaj sınıfları arasında iyi bir ayıraç olmayacağı düşünülmüştür.

Bu aşamadan sonra filtre yeni gördüğü bir mesajı kelimelere ayıracak ve yukarda belirtildiği gibi kelimelere olasılıkları atadıktan sonra Bayes teoremi yoluyla

bu kelimeli olasılıklarını birleştirip elektronik postanın istenmeyen elektronik posta olup olmadığını belirleyecektir.

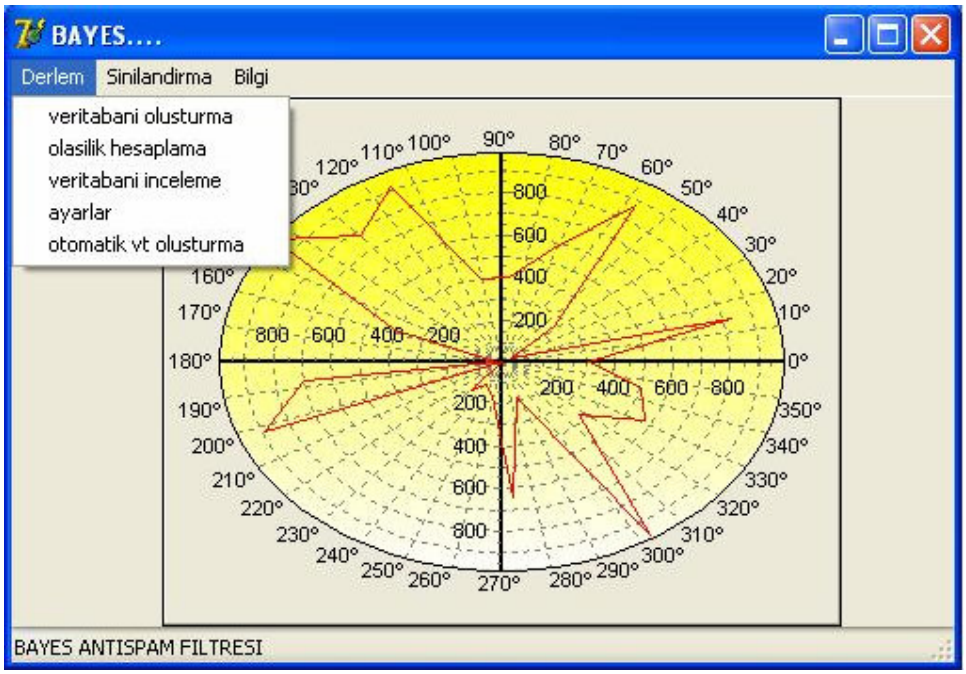
5.5.4 İstatistiksel kombinasyon

Filtrelenecek olan mesajdaki bütün kelimeleri kullanmak yerine sadece olasılıkları 0,5 yani nötr den en uzak olan (0 ve 1 e en yakın olan) toplam 20 adet kelime seçilmiş ve bu kelimelerin olasılıkları Bayes teoremine göre birleştirilmiştir. Bu işlem yalın Bayes öğrenmedeki özellik seçimi ile örtüşmektedir.

a,b,c….n kelimelerimizin daha önceden hesaplanan olasılıkları olmak üzere bu olasılık Bayes teoremi yoluyla aşağıdaki gibi birleştirilerek ilgili elektronik posta için bir skor hesaplanmıştır. Bu skorun 0,9'dan büyük olduğu durumlarda elektronik posta istenmeyen elektronik posta olarak değerlendirilmiştir.

$$p = \frac{a.b.c.....n}{a.b.c.....n + (1-a)(1-b)(1-c).....(1-n)}$$

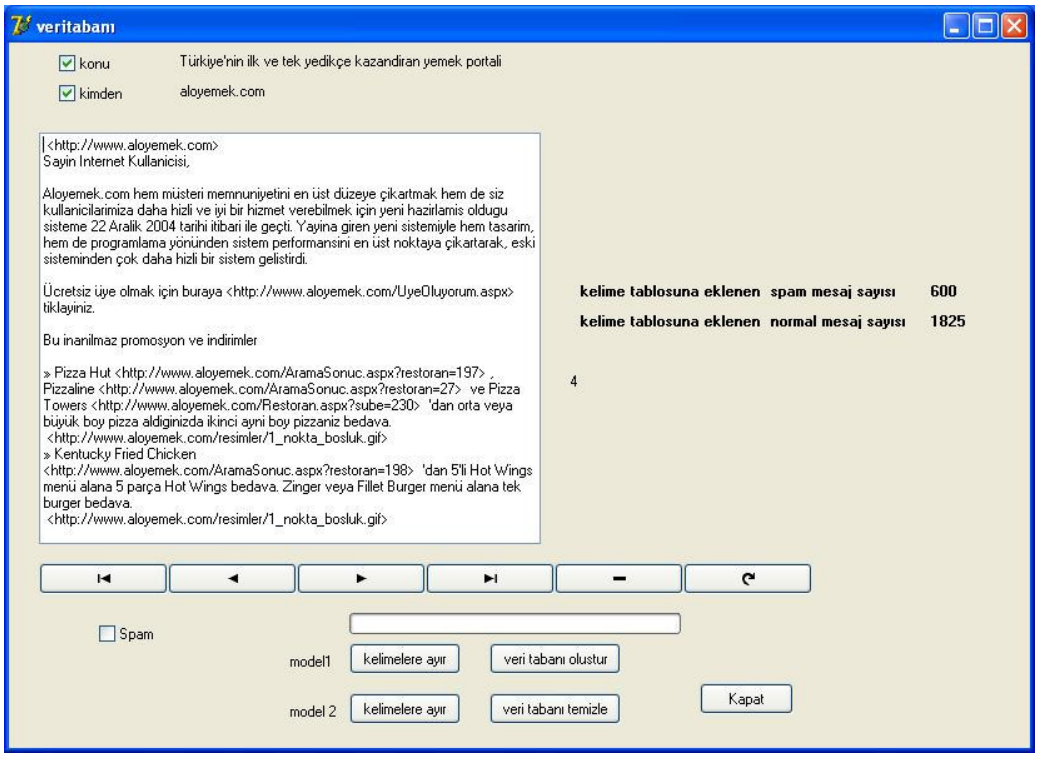
5.5.5 Prototip filtreleme programının tanıtımı



ŞEKİL 3.1 Filtre ana menü

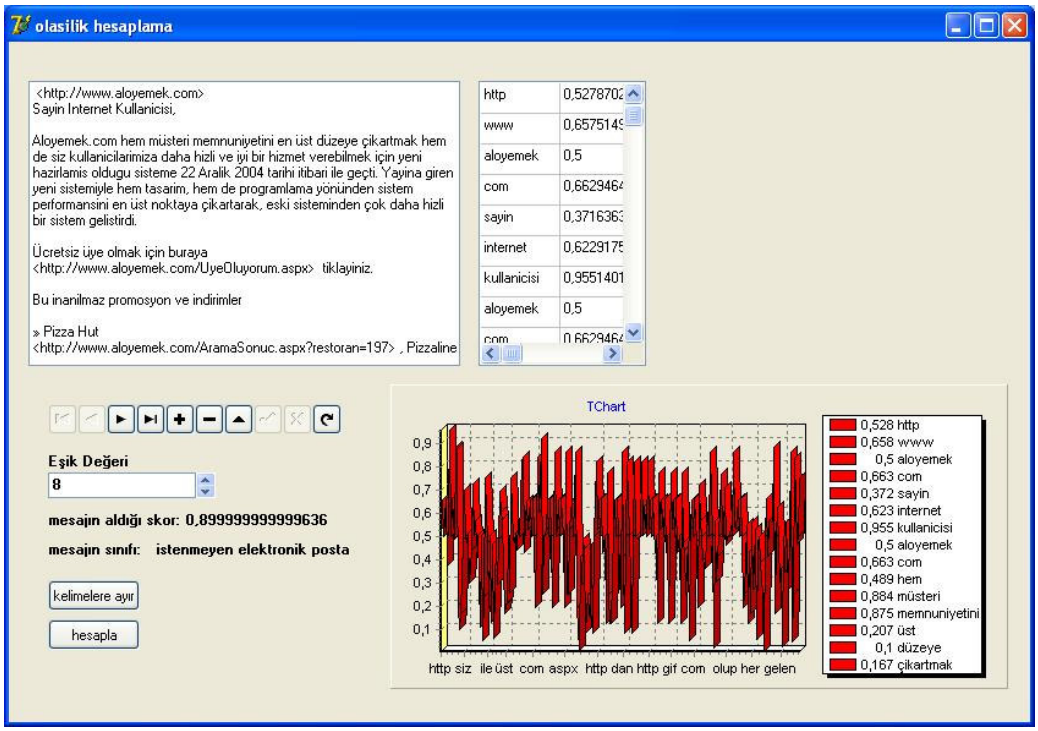
Kullanıcı önce filtrenin öğrenme işlemini gerçekleştirebilmesi için derlem menüsünden bir talim kümesi yani veritabanı oluşturmak zorundadır.

Microsoft Outlook veya Outlook Express programlarından derlemi için depoladığı elektronik postaları Microsoft Access veritabanı formatında kaydetmelidir. Bu aşamadan sonra filtre bu elektronik posta deposunu tanıyabilecektir.



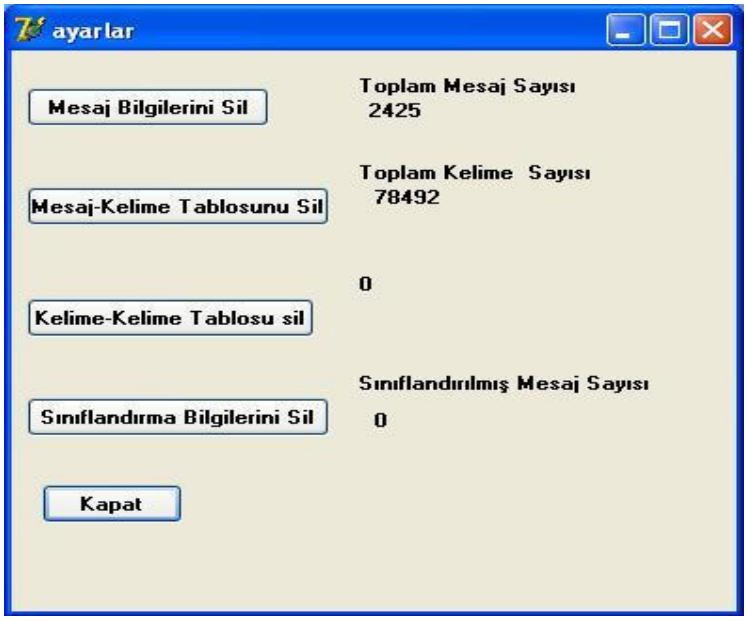
ŞEKİL 3.2 Filtre veritabanı ekranı

Şekil 3.2'de görüldüğü gibi veritabanı kısmında kullanıcı rasgele hazırlanmış bir elektronik posta deposundan, elektronik postaları inceleyerek istenmeyen veya normal olarak kelimelere ayırma ve olasılık atama işlerini daha önce de bahsettiğimiz modellere göre yapabilir, kimden ve konu gibi mesaj üstbilgilerini de dahil edebilir, daha sonra kelime veritabanını oluşturabilir.



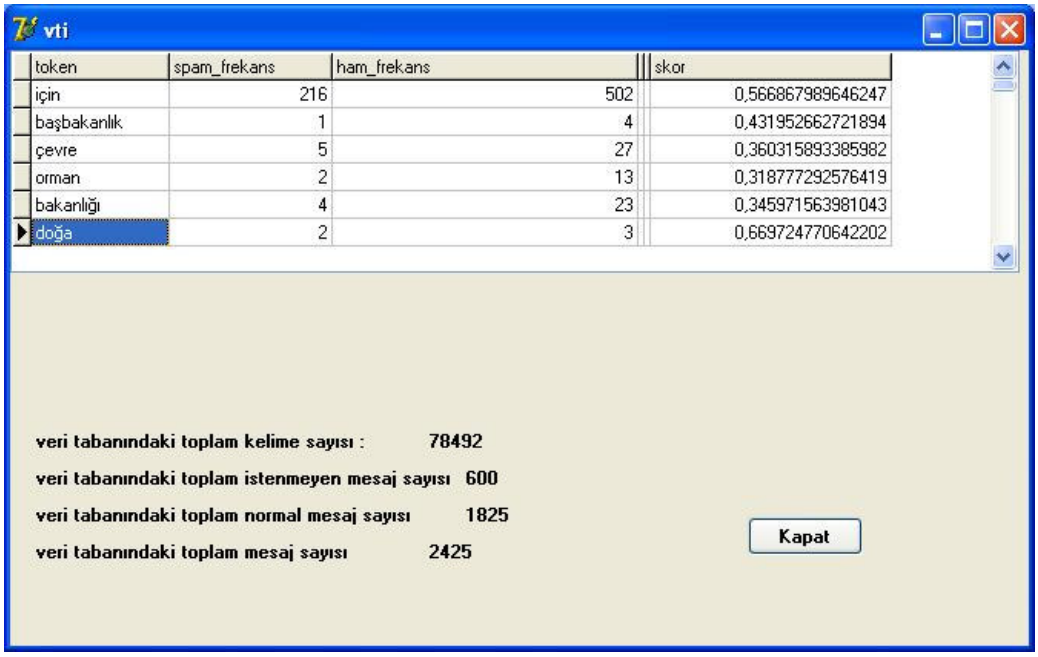
ŞEKİL 3.3 Filtre olasılık hesaplama ekranı

Şekil 3.3'de görüldüğü gibi olasılık hesaplama kısmı filtrenin yaptığı işlemleri aşamalar halinde görmek ve incelemek için tasarlanmıştır.



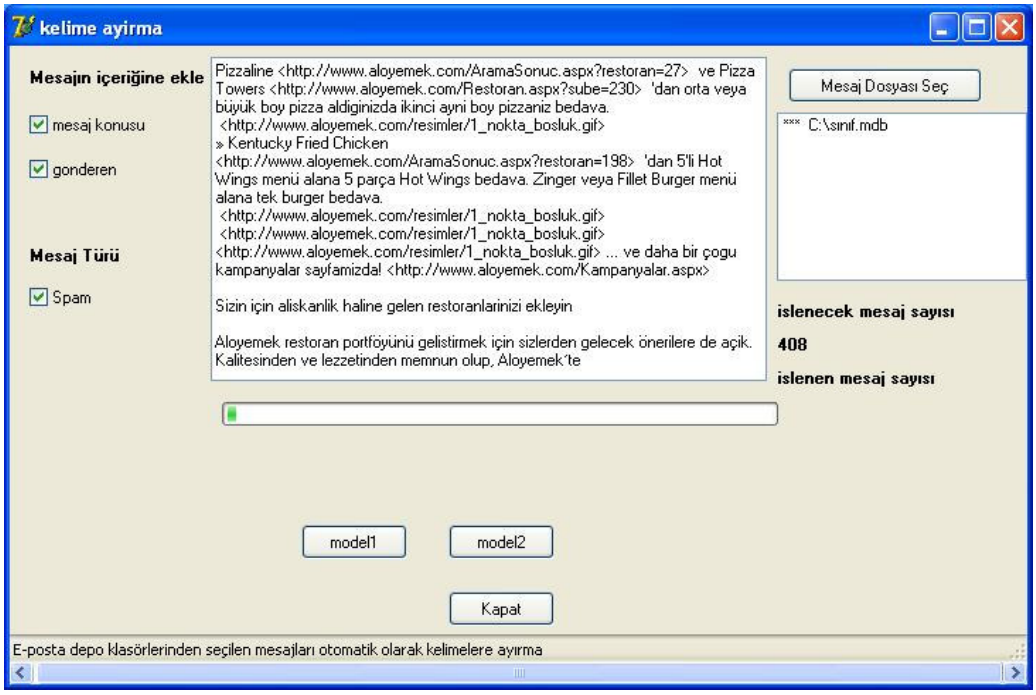
ŞEKİL 3.4 Filtre ayarlar ekranı

Şekil 3.4'de görüldüğü gibi ayarlar kısmında filtrenin talim ve test bilgilerinin silinmesi işlemleri yapılabilmektedir.



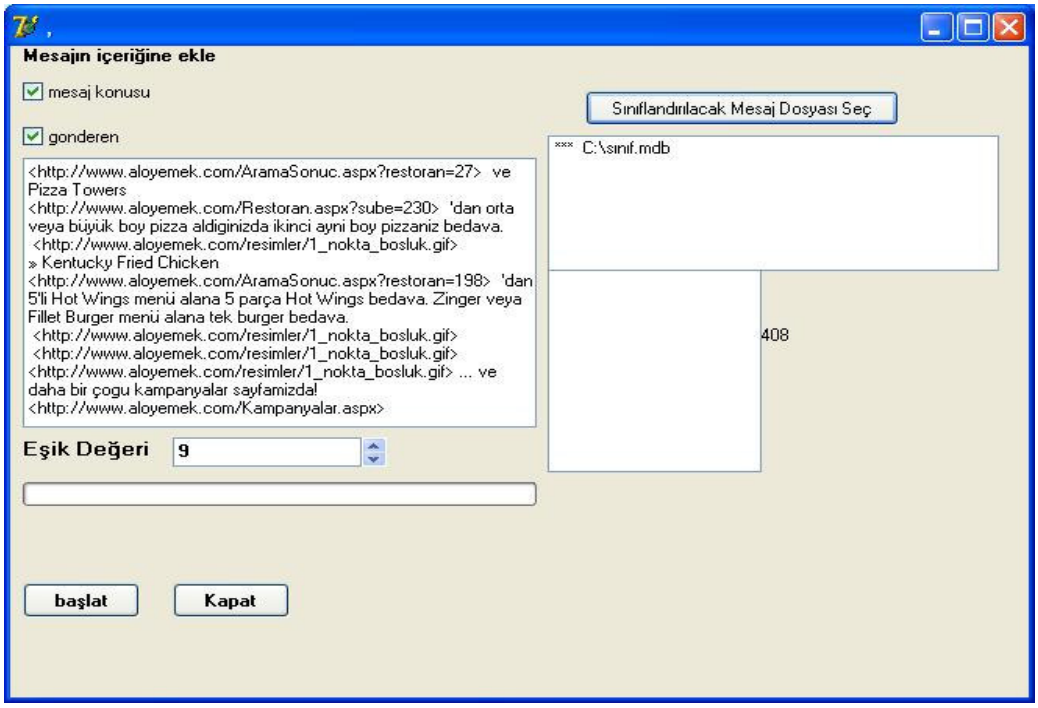
ŞEKİL 3.5 Filtre istatistik ekranı

Şekil 3.5'de görüldüğü gibi inceleme ekranı veritabanıyla ilgili çeşitli istatistikler ve kelimelerin oluş sayıları, olasılıkları incelenebilir.



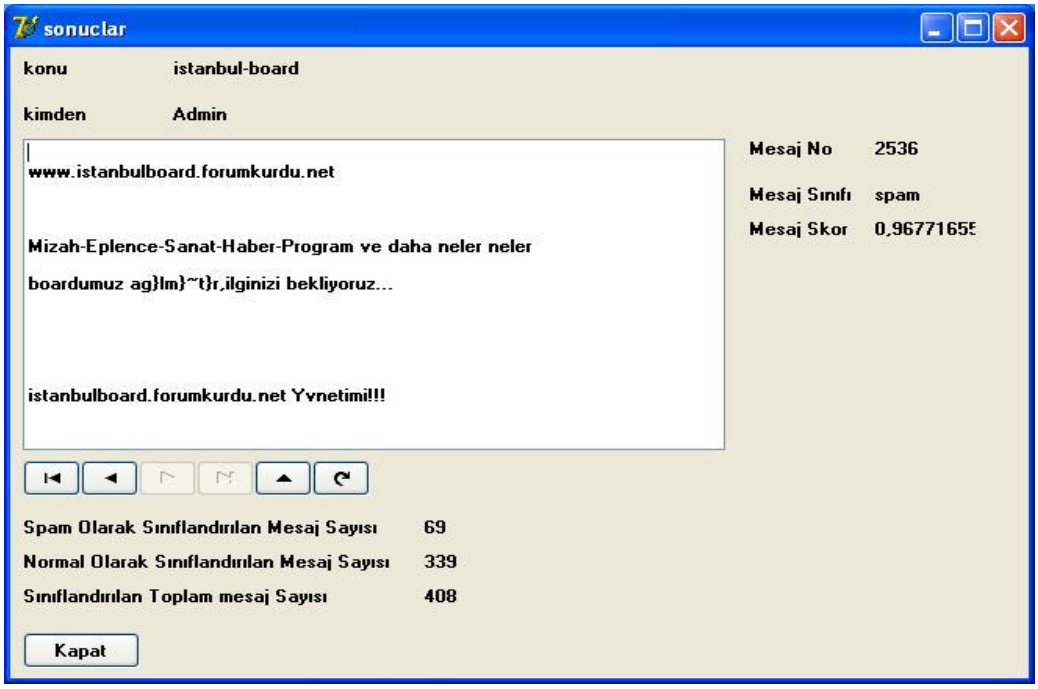
ŞEKİL 3.6 Filtre öğrenme ekranı

Şekil 3.6'da görüldüğü gibi veritabanı kısmındaki işlemin önceden istenmeyen veya normal olduğunu bildiğimiz elektronik posta depolarını seçerek otomatik olarak yapılabilmesini sağlar.



ŞEKİL 3.7 Filtre sınıflandırma ekranı

Şekil 3.7'de görüldüğü gibi sınıflandırma menüsünden kullanıcı sınıflandıracağı mesaj deposunu yani test kümesini belirler.



ŞEKİL 3.8 Filtre analiz ekranı

Şekil 3.8'de görüldüğü gibi sonuçlar menüsünden sınıflandırma sonuçları değerlendirilebilir.

6. ARAŞTIRMA BULGULARI

Ölçüler : Duyarlık, Anma, Doğruluk

Bu çalışmada Filtrelemenin başarısı değerlendirilirken duyarlık, anma ve doğruluk ölçüleri kullanılmıştır. Değerlendirmek istediğimiz şey; filtrenin istenmeyen elektronik posta seçebilme başarısıdır.

TABLO 3.1 Olasılık tablosu

	(Uzman Görüşü) Evet (True)	(Uzman Görüşü) Hayır(False)
İstenmeyen e-posta (positive)	**a**	**b**
Normal e-posta (negative)	**c**	**d**

a (true positive)

b (false positive)

c (true negative)

d (false negative)

a = filtrenin istenmeyen elektronik posta olarak seçtiği ve uzman kişi tarafından istenmeyen elektronik posta olarak belirlenmiş elektronik posta sayısı

b = filtrenin istenmeyen olarak seçtiği fakat uzman kişi tarafından normal olduğu belirlenmiş elektronik posta sayısı

c = filtrenin normal olarak seçtiği ve uzman kişi tarafından normal elektronik posta olarak belirlenen elektronik posta sayısı

d = filtrenin normal olarak seçtiği fakat uzman kişi tarafından istenmeyen olarak belirlenmiş elektronik posta sayısı

Bu bilgiler ışığında istenmeyen elektronik posta içinduyarlık ve anma aşağıdaki gibi hesaplanır.

$$\text{Duyarlık} = \frac{a}{a+b} \qquad \text{Anma} = \frac{c}{c+d}$$

1. Deney

Bu deney için toplam 2387 adet mesajdan oluşan derlemimiz talim ve test kümesi olmak üzere ikiye bölünmüştür. Talim kümesi alınma tarihlerine göre sıralanmış, 667 adet istenmeyen ve 1400 adet normal elektronik posta mesajı olmak üzere 2067 adet elektronik posta mesajından oluşturulmuştur.

Test kümesi ise yine alınma tarihine göre sıralanmış ve alınma tarihleri talim kümesindeki mesajlardan daha sonra olan, 100'ü istenmeyen ve 220'si normal elektronik posta mesajı olmak üzere toplam 320 mesajdan oluşturulmuştur. Test kümesindeki mesajların filtrenin daha önceden hiç görmediği mesajlar olması ölçme işleminin doğruluğunu arttıracaktır.

İlk modelimizi esas ve eşik değerini 0,05 olarak belirlediğimiz filtreleme işleminde;

İstenmeyen elektronik postalar için

$$\text{Duyarlık} = \frac{81}{81+19} = 0{,}81 = \%81$$

$$\text{Anma} = \frac{220}{220+19} = 0{,}92 = \%92$$

Normal elektronik postalar için

$$\text{Duyarlık} = \frac{220}{220+0} = 1{,}00 = \%100$$

$$\text{Anma} = \frac{81}{81+19} = 0{,}81 = \%81$$

olarak elde edilmiştir.

2. Deney

İkinci model ve eşik değer olarak 0,05 belirlenmiş filtreleme işleminde;

İstenmeyen elektronik postalar için

$$\text{Duyarlık} = \frac{84}{84+16} = 0{,}84 = \%84$$

$$\text{Anma} = \frac{220}{220+16} = 0{,}932 = \%93{,}2$$

Normal elektronik postalar için

$$\text{Duyarlık} = \frac{220}{220+0} = 1{,}00 = \%100$$

$$\text{Anma} = \frac{84}{84+16} = 0{,}84 = \%84$$

olarak elde edilmiştir.

Her iki deneyin sonuçları da Tablo 3.2'de verilmiştir.

TABLO 3.2 Karşılaştırmalı olasılık tablosu

	Toplam mesaj sayısı	Test Edilen mesaj sayısı	İstenmeyen elektronik posta mesajları		Normal elektronik posta mesajları	
			Duyarlık (%)	Anma(%)	Duyarlık (%)	Anma (%)
Model 1	2387	320	81	92	100	81
Model 2	2387	320	84	93,2	100	84

Kaynak özetleri kısmında belirtilen çalışmalarda Bayes filtreleme ile %100 e yakın istenmeyen elektronik posta duyarlığı yakalanmış fakat normal elektronik posta için bu değer daha gerilerde kalmıştır. Bunun sebebi olarak kullanılan derlemin istenmeyen ve normal elektronik posta dengesi gösterilebilir. Bu çalışmalarda kullanılan derlemlerde çoğunlukla istenmeyen elektronik posta sayısı normal elektronik posta sayısından fazladır.

Bu çalışmada normal elektronik posta belirlemede duyarlık %100 e ulaşmıştır. Çünkü kullandığımız derlemde normal elektronik posta sayısı istenmeyen elektronik posta sayısının iki katından biraz fazladır ve toplanılan normal elektronik postalar tek bir kullanıcının elektronik posta arşivinden, istenmeyen elektronik postalar ise çok sayıda kullanıcının elektronik posta arşivinden derlenmiştir.

Normal elektronik postalar tek bir kullanıcının arşivinden alındığı için karakteristikleri birbirine çok benzerdir ve filtremiz bu mesajları öğrenmede daha başarılı olmuştur. İstenmeyen elektronik posta mesajları ise farklı kullanıcıların elektronik posta arşivlerinden alındığından karakteristikleri açısından normal ve tek bir kullanıcıdan alınmış elektronik posta mesajlarına oranla daha değişken olma eğilimindedirler. Kelime olasılıklarını hesaplarken kullanılan modeller açısından da kelime oluşlarının da göz önünde bulundurulduğu ikinci modelimiz daha başarılı olmuştur.

7. SONUÇLAR ve TARTIŞMA

Şimdiye kadar yapılan çalışmalarda, gerek yalın Bayes sınıflandırmanın istenmeyen elektronik posta belirleme problemine uygulanışı, gerekse Bayes filtreleme adı altında Bayes istatistiğinin istenmeyen elektronik posta belirleme problemine uygulanışı çeşitli deneylerle incelenmiş ve çok başarılı sonuçlar elde edilmiştir. Bugün Bayes filtreleme ile çalışan çok sayıda hem istemci hem sunucu tabanlı istenmeyen elektronik posta engelleme yazılımı mevcuttur. İstenmeyen elektronik posta engelleme teknikleri bahsinde de belirtildiği gibi bugün bu probleme çok farklı açılardan yaklaşılmaktadır. Bayes filtreleme de diğer tekniklerin yanında vazgeçilmez bir unsurdur. Bununla birlikte sadece Bayes filtreleme ile çalışan açık kodlu yazılımların sayısı zamanla artmakta ve kullandıkları yöntemlerde küçük farklılıklar göstermektedir.

Bu çalışmada kelime oluşlarının sayısını da hesaba katmanın filtreleme başarısını daha da arttırdığı görülmüştür. İstenmeyen ve normal elektronik postaları tek bir kullanıcının arşivinden ve sayıca daha fazla miktarlarda seçmenin başarımı çok daha arttıracağı aşikardır.

Tamamı Türkçe elektronik posta mesajlarından oluşturulan derlem üzerinde iki deney yapılmıştır. Mesajların Türkçe olmasının çalışmayla alakası sadece Türkçe istenmeyen elektronik posta gönderenlerin mesaj karakteristiklerini değerlendirme bağlamındadır. Mesajlarda herhangi bir dilbilimsel çalışma yapılmamıştır. İkili veya üçlü ifadeler (“bedava mp3”, “ücretsiz üyelik” v.s) ayrıca eklenmemiş, sadece kelimelerin bireysel frekansları dikkate alınmıştır. Html içerikli elektronik posta mesajlarında olduğu gibi alınmış herhangi bir dönüşüm yapılmamıştır. Bu hususlarında göz önünde bulundurulmasının sağlayacağı katkılar aşikardır.

Derlemimizde yer alan Türkçe istenmeyen elektronik postalar incelendiğinde, göndericilerinin genellikle yabancı meslektaşlarına oranla daha üşengeç veya basit yöntemler kullandıkları dikkat çekmektedir. Örneğin sıklıkla mesajların sadece konularına bir harf ekleyip çıkararak veya gönderen ismini değiştirerek belli periyotlarda (büyük çoğunlukla günün aynı saatlerinde) bu mesajları göndermektedirler. Kullandıkları ip blokları gene çok büyük yüzdeyle

Kablo Net, ADSL ve Telekom'un Çevirmeli Ağ Bağlantısı internet erişim türlerine atadığı ip adres bloklarındandır. Buradan hareketle bundan sonraki çalışmalarda mesajın gönderildiği tarih, gönderen bilgisayarın ip adresi gibi mesaj üstbilgileri de Bayes filtrelemeye dahil edilebilir. Ayrıca istenmeyen elektronik postaları normal elektronik postalardan büyük ölçüde ayıran bir husus da aynı anda çok sayıda kişiye gönderilmeleridir. Bu durum istenmeyen posta mesajlarının boyutlarıyla ilgili de bize fikir vermektedir. Örneğin bir istenmeyen elektronik posta mesajı herhangi bir ek dosya içerme eğiliminde olmayacaktır. Bir ek dosya içermesi durumunda hem dağıtımı zorlaşacak hem de kullanıcı mesajın konusunu görür görmez ekte iletilen bilgiyi dikkate almayacaktır. Bu tip küçük ayrıntıları da dikkate almak faydalı olacaktır.

KAYNAKLAR

A. McCallum and K. Nigam, 1998. A comparison of event models for Naive Bayes text classification. In AAAI-98 Workshop on Learning for Text Categorization.

Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos, 2000. Learning to filter spam e-mail: A comparison of a naive Bayesian and a memory-based approach. In Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases

Cohen, W. W. ,1996a. Learning rules that classify e-mail. In Papers from the AAAI Spring Symposium on Machine Learning in Information Access, 18--25.

Cormac O'Brien and Carl Vogel , 2003. Spam Filters: Bayes vs. Chi-squared; Letters vs. Words, In *Proceedings of the International Symposium on Information and Communication Technologies*

Graham Paul, 2003. Beter Bayesian Filtering. In Proceedings of the Spam Conference, http://www.paulgraham.com/better.html

H. Drucker, with D. Wu and V.Vapnik, 1999. Support Vector Machines for Spam Categorization.. IEEE Trans. on Neural Networks , vol 10, number 5, pp. 1048-1054.

J. Provost, 1999. Naive-Bayes vs. Rule-Learning in Classification of Email University of Texas at Austin, Artificial Intelligence Lab. Technical Report AI-TR-99-284

J. Rennie, 2000. "*ifile: An application of machine learning to e-mail filtering*," Proc. KDD Workshop on Text Mining, 24

L. Zhang, J. Zhu, T. Yao, 2004. An evaluation of statistical spam filtering techniques. ACM Transactions on Asian Language Information Processing (TALIP) Volume 3 , Issue 4 ,243 - 269

Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz,1998 A Bayesian approach to filtering junk e-mail. In AAAI-98 Workshop on Learning for Text Categorization

X. Carreras and L. Mrquez, 2001. Boosting trees for anti-spam email filtering. In Proceedings of RANLP-01, Jth International Conference on Recent Advances in Natural Language Processing, Tzigov Chark, BG

Y. Diao, H. Lu, D. Wu., 2000. A Comparative Study of Classification Based Personal E-mail Filtering Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications , 408-419

Z. Chuan, L. Xianliang, H. Mengshu, Z. Xu, 2005 A LVQ-based neural network anti-spam email approach. ACM SIGOPS Operating Systems Review. Volume 39, Issue 1 , 34-39

Zdziarski, Jonathan. A, 2005. Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification, No Starch Pres Inc.

Ek 1

```
program Project1;
uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1},
  Unit2 in 'Unit2.pas' {Form2},
  Unit3 in 'Unit3.pas' {Form3},
  Unit4 in '..\..\prog\Unit4.pas' {Form4},
  Unit5 in '..\..\prog\Unit5.pas' {DataModule5: TDataModule},
  Unit6 in 'Unit6.pas' {Frame6: TFrame},
  Unit7 in 'Unit7.pas' {Form7},
  Unit8 in 'Unit8.pas' {Form8},
  Unit9 in 'Unit9.pas' {Form9},
  Unit10 in '..\..\derlemler\Unit10.pas' {DataModule10: TDataModule},
  Unit11 in 'Unit11.pas' {Form11};
{$R *.res}
begin
  Application.Initialize;
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm4, Form4);
  Application.CreateForm(TDataModule5, DataModule5);
  Application.CreateForm(TForm7, Form7);
  Application.CreateForm(TForm8, Form8);
  Application.CreateForm(TForm9, Form9);
  Application.CreateForm(TDataModule10, DataModule10);
  Application.CreateForm(TForm11, Form11);
  Application.Run;
end.
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, StdCtrls, DBCtrls, ADODB, ExtCtrls, Mask, Menus, OleServer,
  OutlookXP, Grids, ComCtrls, TeEngine, Series, TeeProcs, Chart, StatChar,
  CurvFitt, TeeFunci, Unit6;
type
  TForm1 = class(TForm)
    DBMemo1: TDBMemo;
    DBNavigator1: TDBNavigator;
    DBText1: TDBText;
    DBText2: TDBText;
    DBText3: TDBText;
    Label1: TLabel;
    Label2: TLabel;
```

```
    Label3: TLabel;
    Button5: TButton;
    Button7: TButton;
    StringGrid1: TStringGrid;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Edit1: TEdit;
    Button8: TButton;
    Label7: TLabel;
    Label9: TLabel;
    Label8: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    UpDown1: TUpDown;
    Chart1: TChart;
    Series1: TLineSeries;
    Button1: TButton;
    function getXY(x:integer ;S :String):integer;
    function getXY2(x:integer ;S :String):integer;
    procedure Button1Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button8Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  c:integer;
 implementation

uses Unit2, Unit5;
{$R *.dfm}

function TForm1.getXY(x:integer ;s :String):integer;
var
i:integer;
begin
Result:=0;
for i:=x to length(s) do begin
if ((s[i] in  ['a'..'z','ç','ğ','ş','ı','ö','ü','A'..'Z','Ç','Ğ','İ','Ö','Ü','Ş'])) then
begin
```

```
   Result:=i;
   break;
end;
end;
if Result=0 then Result:=i-1;
end;
function TForm1.getXY2(x:integer ;s :String):integer;
var
i:integer;
begin
Result:=0;
for i:=x to length(s)+1 do begin
if (not(s[i] in  ['a'..'z','ç','ğ','ş','ı','ö','ü','A'..'Z','Ç','Ğ','İ','Ö','Ü','Ş'])) then
begin
   Result:=i;
   break;
end;
if Result=0 then Result:=i+1;
end;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
form1.Close;
end;
procedure TForm1.Button5Click(Sender: TObject);
var
z,i,bas,n,bit,p,x,y,cc,nn:integer;
f,s:string;
d:char;
begin
DataModule5.derlem.Close;
DataModule5.derlem.Open;
i:=0;
y:=0;
p:=0;
z:=0;
 n:=1;
 nn:=1;
for z:=0 to DBMemo1.Lines.Count-1 do begin
i:=0;
f:=DBMemo1.Lines[z];
if (length(f)=0) then continue;
repeat
x:=getXY(i,f);
if (x=length(f)) then break;
y:=getXY2(x+1,f);
s:=AnsiLowerCase(copy(f,x,y-x));
DataModule5.derlem.Locate('token',s,[]);
```

```
if DataModule5.derlemtoken.AsString=s then
  begin
       StringGrid1.Cells[1,n]:=floattostr(DataModule5.derlemskor.AsFloat);
        if DataModule5.derlemspam_frekans.AsInteger=0 then

StringGrid1.Cells[1,n]:=floattostr(0.5/(DataModule5.derlemham_frekans.asinteger+
1));
        if DataModule5.derlemham_frekans.asinteger=0 then
        StringGrid1.Cells[1,n]:=floattostr(1-
(0.5/(DataModule5.derlemspam_frekans.asinteger+1)));

        end
        else

        begin
        StringGrid1.Cells[1,n]:=floattostr(0.5);
        end;

if (length(s)>2) then begin
StringGrid1.Cells[0,n]:=s;
n:=n+1;
StringGrid1.RowCount:=n;
   end;
i:=y+1;
until (i>=length(f))
end;
end;
procedure TForm1.Button7Click(Sender: TObject);
var
z,t,h:Real48;
dn:byte;
begin
for dn:=1 to StringGrid1.RowCount-1 do begin
Series1.AddY(strtofloat(StringGrid1.Cells[1,dn]),(StringGrid1.Cells[0,dn]));
z:=strtofloat(StringGrid1.Cells[1,dn]);
t:=1-z;
t:=t*t;
z:=z*z;
h:=(z/(z+t));
end;
label4.Visible:=true;
Label4.Caption:=floattostr(h);
Label5.Visible:=true;
if h>=strtofloat(edit1.Text) then
label5.caption:='mesaj spamdır' else
label5.Caption:='mesaj normaldir';
end;
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
series1.Clear;
end;
procedure TForm1.Button8Click(Sender: TObject);
var s,ss,k,v,ccc:double;
g,n:integer;
begin
s:=0;
k:=0;
v:=0;
ccc:=0;
for g:=1 to StringGrid1.RowCount-1 do begin
s:=s+strtofloat(StringGrid1.Cells[1,g]);
end;
v:=s/(StringGrid1.rowcount-1);
for n:=1 to StringGrid1.RowCount-1 do begin
ccc:=StrToFloat(stringgrid1.Cells[1,n]);
ccc:=(ccc-v)*(ccc-v);
k:=k+ccc;
end;
ss:=sqrt(k/(StringGrid1.RowCount-2));
Label7.Visible:=true;
Label9.visible:=true;
Label7.Caption:=floattostr(ss);
Label9.Caption:=floattostr(v);
end;

procedure TForm1.FormShow(Sender: TObject);
begin
DataModule5.ADOConnection1.Close;
DataModule5.ADOConnection1.Open;
end;
end.
unit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DBCtrls, Menus, XPMan, ComCtrls, TeEngine, Series, TeePolar,
  ExtCtrls, TeeProcs, Chart, StdCtrls;
type
  TForm2 = class(TForm)
    MainMenu1: TMainMenu;
    basla1: TMenuItem;
    veritabani1: TMenuItem;
    sinilandir1: TMenuItem;
    sonuclar1: TMenuItem;
    cikis1: TMenuItem;
    olasilikhesapla1: TMenuItem;
```

```
    sonuclar2: TMenuItem;
    veritabaniinceleme1: TMenuItem;
    mesajbilgilerisil1: TMenuItem;
    otomatikvtolusturma1: TMenuItem;
    XPManifest1: TXPManifest;
    StatusBar1: TStatusBar;
    k1: TMenuItem;
    Chart1: TChart;
    Series1: TPolarSeries;
    procedure veritabani1Click(Sender: TObject);
    procedure siniflandir1Click(Sender: TObject);
    procedure olasilikhesapla1Click(Sender: TObject);
    procedure veritabaniinceleme1Click(Sender: TObject);
    procedure sonuclar2Click(Sender: TObject);
    procedure sonuclar1Click(Sender: TObject);
    procedure mesajbilgilerisil1Click(Sender: TObject);
    procedure otomatikvtolusturma1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure k1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form2: TForm2;
implementation
uses Unit1, Unit3, Unit4, Unit7, Unit8, Unit5, Unit9, Unit10, Unit11;
{$R *.dfm}
procedure TForm2.veritabani1Click(Sender: TObject);
begin
form4.ShowModal;
end;

procedure TForm2.siniflandir1Click(Sender: TObject);
begin
form3.showmodal;
end;
procedure TForm2.olasilikhesapla1Click(Sender: TObject);
begin
Form1.ShowModal;
end;

procedure TForm2.veritabaniinceleme1Click(Sender: TObject);
begin
form7.showmodal;
end;
procedure TForm2.sonuclar2Click(Sender: TObject);
```

```
begin
form8.showmodal;
end;
procedure TForm2.sonuclar1Click(Sender: TObject);
begin
form3.ShowModal;
end;
procedure TForm2.mesajbilgilerisil1Click(Sender: TObject);
begin
form9.ShowModal;
end;
procedure TForm2.otomatikvtolusturma1Click(Sender: TObject);
begin
form11.showmodal;
end;
procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
free;
DataModule5.ADOConnection1.Connected:=false;
DataModule10.ADOConnection1.Connected:=false;
end;

procedure TForm2.k1Click(Sender: TObject);
begin
form2.Close;
halt;
end;
end.
unit Unit3;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ADODB, ExtCtrls, DBCtrls, Grids, ComCtrls, XPMan;

type
  TForm3 = class(TForm)
    DBNavigator1: TDBNavigator;
    Memo1: TMemo;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    UpDown1: TUpDown;
    Edit1: TEdit;
    Label1: TLabel;
    XPManifest1: TXPManifest;
    StringGrid2: TStringGrid;
    Button2: TButton;
    Label2: TLabel;
    Button3: TButton;
```

```
    ProgressBar1: TProgressBar;
    Label4: TLabel;
    Button1: TButton;
    OpenDialog1: TOpenDialog;
    ADOTable1: TADOTable;
    ListBox1: TListBox;
    Label3: TLabel;
    Label5: TLabel;
    ADOTable1Konu: TWideStringField;
    ADOTable1kimden: TWideStringField;
    ADOTable1SenderName: TWideStringField;
    ADOTable1To: TWideStringField;
    ADOTable1MessageSize: TIntegerField;
    ADOTable1icerik: TMemoField;
    ADOTable1m_no: TAutoIncField;
    ListBox2: TListBox;
    procedure Button1Click(Sender: TObject);
     function getXY(x:integer ;S :String):integer;
    function getXY2(x:integer ;S :String):integer;
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
  procedure memoya_at();
  procedure hesapla();
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form3: TForm3;
implementation
uses Unit5, Unit10;

{$R *.dfm}
function TForm3.getXY(x:integer ;s :String):integer;
var
i:integer;
begin
Result:=0;
for i:=x to length(s) do begin
if ((s[i] in  ['a'..'z','ç','ğ','ş','ı','ö','ü','A'..'Z','Ç','Ğ','İ','Ö','Ü','Ş'])) then
begin
    Result:=i;
    break;
end;
end;
if Result=0 then Result:=i-1;
```

```
end;
function TForm3.getXY2(x:integer ;s :String):integer;
var
i:integer;
begin
Result:=0;
for i:=x to length(s)+1 do begin
if (not(s[i] in  ['a'..'z','ç','ğ','ş','ı','ö','ü','A'..'Z','Ç','Ğ','İ','Ö','Ü','Ş'])) then
begin
   Result:=i;
   break;
end;
if Result=0 then Result:=i+1;
end;
end;

procedure TForm3.memoya_at();
begin
memo1.Lines.Add(ADOTable1icerik.AsVariant);
if CheckBox1.Checked=true then
Memo1.Lines.Add(ADOTable1Konu.AsString);
if CheckBox2.Checked=true then
Memo1.Lines.Add(adotable1kimden.AsString);
end;
procedure tform3.hesapla;
var
z,t,h:double;
dn:integer;
begin
for dn:=1 to StringGrid2.RowCount-1 do begin
z:=strtofloat(StringGrid2.Cells[1,dn]);
t:=1-z;
t:=t*t;
z:=z*z;
h:=(z/(z+t));
end;
 datamodule5.sinif.Close;
 datamodule5.sinif.Open;
 datamodule5.sinif.Edit;
 datamodule5.sinifmesajskor.AsFloat:=h;
if h>=(strtoint(edit1.Text))/10 then begin
datamodule5.sinifmesaj_sinif.asstring:='spam';
datamodule5.sinifmes_no.AsInteger:=datamodule5.iletilerm_no.AsInteger;
end
 else begin
datamodule5.sinifmesaj_sinif.AsString:='normal';
datamodule5.sinifmes_no.AsInteger:=datamodule5.iletilerm_no.AsInteger;
end;
```

```
datamodule5.sinif.Post;
 end;
  procedure TForm3.Button1Click(Sender: TObject);
  var
s:string;
begin
OpenDialog1.Execute;
s:=OpenDialog1.FileName;
ADOTable1.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;Data
Source='+s+';'+'Persist Security Info=False';
ADoTable1.Active:=true;
ListBox1.Items.Add('***  '+s);
label5.Caption:='';
label5.Caption:=inttostr(adotable1.RecordCount);
ProgressBar1.Max:=adotable1.RecordCount;
ProgressBar1.Position:=0;
end;

procedure TForm3.Button2Click(Sender: TObject);
var
z,i,bas,n,bit,p,x,y,cc,nn,st,say,ss,dn,zzz:integer;
zz,t,h,dns,k,u:extended;
f,s:string;
 d:char;
label duzelt;
begin

ADOTable1.Open;
say:=0;
ADOTable1.DisableControls;
adotable1.First;
while not adotable1.Eof do begin
ProgressBar1.StepBy(1);
Memo1.Text:='';
memoya_at;
ss:=0; i:=0; y:=0; p:=0; z:=0; n:=1; nn:=1;  dn:=0;    zzz:=1;
 k:=1;    zz:=0;
 u:=1;
for z:=0 to Memo1.Lines.Count-1 do begin
i:=0;
f:=Memo1.Lines[z];
if (length(f)=0) then continue;
repeat
x:=getXY(i,f);
if (x=length(f)) then break;
y:=getXY2(x+1,f);
s:=AnsiLowerCase(copy(f,x,y-x));
```

```
DataModule5.derlem.Locate('token',s,[]);
if DataModule5.derlemtoken.AsString=s then
  begin
      if
(DataModule5.derlemham_frekans.AsInteger+DataModule5.derlemspam_frekans.As
Integer)<5 then
       goto duzelt;
      StringGrid2.Cells[1,n]:=floattostr(DataModule5.derlemskor.AsFloat);
       if DataModule5.derlemspam_frekans.AsInteger=0 then

StringGrid2.Cells[1,n]:=floattostr(0.5/(DataModule5.derlemham_frekans.asinteger+
1));
       if DataModule5.derlemham_frekans.asinteger=0 then
       StringGrid2.Cells[1,n]:=floattostr(1-
(0.5/(DataModule5.derlemspam_frekans.asinteger+1)));
       end
       else
        begin
       StringGrid2.Cells[1,n]:=floattostr(0.5);
      end;
 if (length(s)>2) then begin
StringGrid2.Cells[0,n]:=s;
n:=n+1;
StringGrid2.RowCount:=n;
  end;
  duzelt:
i:=y+1;
until (i>=length(f))
end;
//hesaplama

if (StringGrid2.RowCount)<=26 then begin
for dn:=1 to (StringGrid2.RowCount-1) do begin
zz:=strtofloat(StringGrid2.Cells[1,dn]);
k:=k*zz;
t:=1-zz;
u:=u*t;
h:=(k/(k+u));
end;
end;
if (StringGrid2.RowCount)>20 then begin
for dn:=1 to (StringGrid2.RowCount-1) do begin
ListBox2.Items.Add(StringGrid2.cells[1,dn]);
end;
ListBox2.Sorted:=true;
ListBox2.Refresh;
zzz:=StringGrid2.RowCount-2;
for ss:=1 to 20 do begin
```

```
if ss<=10 then  begin
zz:=strtofloat(ListBox2.Items[ss-1]);
k:=k*zz;
t:=1-zz;
u:=u*t;
end;
if ss>10 then
begin
zz:=strtofloat(ListBox2.Items[zzz]);
k:=k*zz;
t:=1-zz;
u:=u*t;
zzz:=zzz-1;
end;
t:=0;
end;
h:=(k/(k+u));
end;
 datamodule5.sinif.Close;
 datamodule5.sinif.Open;
 datamodule5.sinif.insert;
 datamodule5.sinifmesajskor.AsFloat:=h;
if h>=(strtoint(edit1.Text))/100 then begin
datamodule5.sinifmesaj_sinif.asstring:='spam';
datamodule5.sinifmes_no.AsInteger:=ADOTable1m_no.AsInteger;
datamodule5.sinif.Post;
end
 else begin
datamodule5.sinifmesaj_sinif.AsString:='normal';
datamodule5.sinifmes_no.AsInteger:=ADOTable1m_no.AsInteger;
datamodule5.sinif.Post;
end;
while st<stringGrid2.Rowcount do
begin
StringGrid2.Rows[st].Clear;
inc(st);
end;
listbox2.Clear;
DataModule5.sinif.Edit;
DataModule5.sinifmesaj_konu.AsString:=ADOTable1Konu.AsString;
DataModule5.sinifmesaj_kimden.AsString:=ADOTable1kimden.AsString;
DataModule5.sinifmesaj_govde.AsVariant:=ADOTable1icerik.AsVariant;
DataModule5.sinif.Post;

adotable1.Next;
say:=say+1;
end;
adotable1.enableControls;
```

```
label4.Visible:=true;
label4.Caption:=inttostr(say)+'adet mesaj sınıflandırılmıştır!!!';
// bitiş
end;
procedure TForm3.Button3Click(Sender: TObject);
begin
form3.Close;
end;
procedure TForm3.FormShow(Sender: TObject);
begin
DataModule5.ADOConnection1.Close;
DataModule5.ADOConnection1.Open;
label5.Caption:='';
label4.Visible:=false;
DataModule5.derlem.Close;
DataModule5.derlem.Open;
end;
end.
unit Unit5;
interface
uses
  SysUtils, Classes, DB, ADODB;
type
  TDataModule5 = class(TDataModule)
    iletiler: TADOTable;
    kelimeler: TADOTable;
    mesaj: TADOTable;
    sinif: TADOTable;
    derlem: TADOTable;
    ADOConnection1: TADOConnection;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    ornek: TADOTable;
    veritabaniyarat: TADOQuery;
    veritabanisil: TADOQuery;
    kelimesil: TADOQuery;
    DataSource3: TDataSource;
    toplamspam: TADOQuery;
    mesajmesaj_no: TAutoIncField;
    mesajmesaj_sinifi: TWideStringField;
    toplamnormal: TADOQuery;
    DataSource4: TDataSource;
    DataSource5: TDataSource;
    DataSource6: TDataSource;
    DataSource7: TDataSource;
    sinifspam: TADOQuery;
    sinifnormal: TADOQuery;
    DataSource8: TDataSource;
```

```
    sinifsil: TADOQuery;
    kelimelertoken: TWideStringField;
    kelimelerspam_frekans: TIntegerField;
    kelimelerham_frekans: TIntegerField;
    kelimelerskor: TIntegerField;
    kelimelermes_no: TIntegerField;
    kelimelerre_no: TAutoIncField;
    mesajsil: TADOQuery;
    derlemtoken: TWideStringField;
    derlemspam_frekans: TIntegerField;
    derlemham_frekans: TIntegerField;
    derlemspamolasilik: TFloatField;
    derlemhamolasilik: TFloatField;
    derlemskor: TFloatField;
    derlemtno: TAutoIncField;
    sinifmes_no: TIntegerField;
    sinifmesaj_sinif: TWideStringField;
    sinifmesajskor: TFloatField;
    sinifmesaj_govde: TMemoField;
    sinifmesaj_kimdenad: TWideStringField;
    sinifmesaj_kimden: TWideStringField;
    sinifmesaj_konu: TWideStringField;
    sinifrsn: TAutoIncField;
    kelimeler2: TADOTable;
    kelimeler2token: TWideStringField;
    kelimeler2spam_frekans: TIntegerField;
    kelimeler2ham_frekans: TIntegerField;
    kelimeler2skor: TIntegerField;
    kelimeler2mes_no: TIntegerField;
    kelimeler2re_no: TAutoIncField;
    kbazlvtyarat: TADOQuery;
    kelime2sil: TADOQuery;
    DataSource9: TDataSource;
    iletilerKonu: TWideStringField;
    iletilericerik: TMemoField;
    iletilerkimden_ad: TWideStringField;
    iletilerkimden: TWideStringField;
    iletilerm_no: TAutoIncField;
    procedure iletilerAfterScroll(DataSet: TDataSet);
    procedure DataModuleCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  DataModule5: TDataModule5;
implementation
```

```
uses Unit4, Unit1;

{$R *.dfm}

procedure TDataModule5.iletilerAfterScroll(DataSet: TDataSet);
begin
form4.Label1.visible:=false;
form4.label2.visible:=false;
form4.ProgressBar1.Position:=0;
end;
end.
unit Unit6;

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs;
type
  TFrame6 = class(TFrame)
  private
    { Private declarations }
  public
    { Public declarations }
  end;
implementation
{$R *.dfm}
end.
unit Unit7;
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, DBGrids, DBCtrls;

type
  TForm7 = class(TForm)
    DBGrid1: TDBGrid;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label8: TLabel;
    DBText1: TDBText;
    DBText2: TDBText;
    Button1: TButton;
    procedure FormShow(Sender: TObject);
    procedure Button1Click(Sender: TObject);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form7: TForm7;
implementation
uses Unit5, Math;
{$R *.dfm}
procedure TForm7.FormShow(Sender: TObject);
begin
DataModule5.ADOConnection1.Close;
DataModule5.ADOConnection1.Open;
DBText1.Visible:=false;
dbtext2.Visible:=false;
Label5.Visible:=false;
label8.Visible:=false;
DataModule5.derlem.Open;
DataModule5.kelimeler.Open;
DataModule5.kelimeler2.open;
DataModule5.mesaj.Open;
if DataModule5.derlem.RecordCount<>0 then
begin
DBText1.Visible:=true;
dbtext2.Visible:=true;
label5.Visible:=true;
label8.Visible:=true;
If DataModule5.kelimeler.RecordCount<>0 then
begin
DataModule5.toplamspam.open;
DataModule5.toplamnormal.open;
 end;
 Label5.Caption:=IntToStr(DataModule5.derlem.RecordCount);
 label8.Caption:=inttostr(DataModule5.mesaj.RecordCount);
 end;
 end;

procedure TForm7.Button1Click(Sender: TObject);
begin
form7.Close;
end;
end.
unit Unit8;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```

```
 Dialogs, StdCtrls, DBCtrls, ExtCtrls;
type
  TForm8 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    DBText1: TDBText;
    DBText2: TDBText;
    Label4: TLabel;
    DBMemo1: TDBMemo;
    DBNavigator1: TDBNavigator;
    DBText3: TDBText;
    DBText4: TDBText;
    Label5: TLabel;
    Label6: TLabel;
    DBText6: TDBText;
    DBText7: TDBText;
    Label8: TLabel;
    Label9: TLabel;
    Button1: TButton;
    Label10: TLabel;
    DBText8: TDBText;
    procedure FormShow(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form8: TForm8;
implementation
uses Unit5;
{$R *.dfm}
procedure TForm8.FormShow(Sender: TObject);
begin
DataModule5.ADOConnection1.Close;
DataModule5.ADOConnection1.Open;
DataModule5.sinif.Open;
DataModule5.sinifspam.close;
DataModule5.sinifspam.open;
DataModule5.sinifnormal.close;
DataModule5.sinifnormal.open;
label4.Caption:=inttostr(DataModule5.sinif.RecordCount);
end;
procedure TForm8.Button1Click(Sender: TObject);
begin
form8.Close;
```

```
end;
end.
unit Unit9;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls;
type
  TForm9 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Button2: TButton;
    Label3: TLabel;
    Label4: TLabel;
    Button3: TButton;
    Button4: TButton;
    Label5: TLabel;
    Label6: TLabel;
    Button5: TButton;
    Label7: TLabel;
    Label8: TLabel;
    procedure FormShow(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form9: TForm9;
implementation

uses Unit5, Unit10;
{$R *.dfm}
procedure TForm9.FormShow(Sender: TObject);
begin
DataModule5.ADOConnection1.Close;
DataModule5.ADOConnection1.Open;
DataModule5.mesaj.Open;
DataModule5.kelimeler.Open;
DataModule5.kelimeler2.open;
DataModule5.sinif.Open;
```

```
label2.Caption:=inttostr(DataModule5.mesaj.RecordCount);
label4.Caption:=inttostr(DataModule5.kelimeler.RecordCount);
 label6.Caption:=inttostr(DataModule5.sinif.RecordCount);
 label7.Caption:=inttostr(DataModule5.kelimeler2.RecordCount);
end;
procedure TForm9.Button1Click(Sender: TObject);
begin
DataModule5.mesajsil.ExecSQL;
end;
procedure TForm9.Button2Click(Sender: TObject);
begin
DataModule5.kelimesil.ExecSQL;
end;
procedure TForm9.FormClose(Sender: TObject; var Action: TCloseAction);
begin
DataModule5.kelimeler.close;
DataModule5.mesaj.close;
DataModule5.sinif.Close;
DataModule5.sinifspam.Close;
DataModule5.sinifnormal.Close;
DataModule5.kelimeler2.close;
end;

procedure TForm9.Button3Click(Sender: TObject);
begin
form9.close;
end;
procedure TForm9.Button4Click(Sender: TObject);
begin
datamodule5.sinifsil.ExecSQL;
ShowMessage('sınıflandırma bilgisi silinmiştir');
end;
procedure TForm9.Button5Click(Sender: TObject);
begin
DataModule5.kelime2sil.ExecSQL;
end;
end.
unit Unit10;
interface
uses
  SysUtils, Classes, DB, ADODB;
type
  TDataModule10 = class(TDataModule)
    ADOConnection1: TADOConnection;
    siniflandirilacak: TADOTable;
    DataSource1: TDataSource;
    siniflandirilacakKonu: TWideStringField;
    siniflandirilacakicerik: TMemoField;
```

```
    siniflandirilacakkimden_ad: TWideStringField;
    siniflandirilacakkimden: TWideStringField;
    siniflandirilacakm_no: TAutoIncField;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  DataModule10: TDataModule10;
implementation
{$R *.dfm}
end.
unit Unit11;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DBCtrls, ComCtrls, ExtCtrls, XPMan, DB, ADODB;
type
  TForm11 = class(TForm)
    Button1: TButton;
    Label2: TLabel;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox4: TCheckBox;
    Button2: TButton;
    XPManifest1: TXPManifest;
    ProgressBar1: TProgressBar;
    Label1: TLabel;
    Button3: TButton;
    Label3: TLabel;
    Button4: TButton;
    StatusBar1: TStatusBar;
    Memo1: TMemo;
    ListBox1: TListBox;
    OpenDialog1: TOpenDialog;
    ADOTable1: TADOTable;
    Label4: TLabel;
    Label5: TLabel;
    Label7: TLabel;
    Label6: TLabel;
    Label8: TLabel;
    ADOTable1Konu: TWideStringField;
    ADOTable1kimden: TWideStringField;
    ADOTable1SenderName: TWideStringField;
    ADOTable1Received: TDateTimeField;
    ADOTable1MessageSize: TIntegerField;
    ADOTable1icerik: TMemoField;
```

```
    ADOTable1m_no: TAutoIncField;
    procedure Button1Click(Sender: TObject);
    function getXY(x:integer ;S :String):integer;
    function getXY2(x:integer ;S :String):integer;
    procedure Button2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form11: TForm11;
implementation
uses Unit5, Unit3, Unit10;
{$R *.dfm}
function TForm11.getXY(x:integer ;s :String):integer;
var
i:integer;
begin
Result:=0;
for i:=x to length(s) do begin
if ((s[i] in  ['a'..'z','ç','ğ','ş','ı','ö','ü','A'..'Z','Ç','Ğ','İ','Ö','Ü','Ş'])) then
begin
    Result:=i;
    break;
end;
end;
if Result=0 then Result:=i-1;
end;
function TForm11.getXY2(x:integer ;s :String):integer;
var
i:integer;
begin
Result:=0;
for i:=x to length(s)+1 do begin
if (not(s[i] in  ['a'..'z','ç','ğ','ş','ı','ö','ü','A'..'Z','Ç','Ğ','İ','Ö','Ü','Ş'])) then
begin
    Result:=i;
    break;
end;
if Result=0 then Result:=i+1;
end;
end;

procedure TForm11.Button1Click(Sender: TObject);
```

```
var
z,i,bas,n,bit,p,x,y,cc,nn,st,say,ff:integer;

zz,t,h:double;
dn:byte;
f,s:string;
d:char;
begin
say:=0;
// burda
adotable1.DisableControls;
adotable1.First;
DataModule5.kelimeler.Close;
DataModule5.kelimeler.Open;
try
while not adotable1.Eof do begin
ProgressBar1.StepBy(1);
Memo1.Text:='';
memo1.Lines.Add(adotable1icerik.AsVariant);
if CheckBox1.Checked=true then Memo1.Lines.Add(adotable1Konu.AsString);
if CheckBox2.Checked=true then Memo1.Lines.Add(adotable1kimden.AsString);

i:=0;
y:=0;
z:=0;

DataModule5.mesaj.append;
if CheckBox4.Checked=true then DataModule5.mesajmesaj_sinifi.AsString:='spam'
else
 DataModule5.mesajmesaj_sinifi.asstring:='normal';
DataModule5.mesaj.Post;

for z:=0 to Memo1.Lines.Count-1 do begin
i:=0;
f:=Memo1.Lines[z];
if (length(f)=0) then continue;
repeat
x:=getXY(i,f);
if (x=length(f)) then break;
y:=getXY2(x+1,f);
s:=AnsiLowerCase(copy(f,x,y-x));
nn:=0;
DataModule5.kelimeler.Locate('token',s,[]);
if (DataModule5.kelimelertoken.AsString=s) then
begin
if
DataModule5.kelimelermes_no.AsInteger<>DataModule5.mesajmesaj_no.AsInteger
then
```

```
  begin
        if CheckBox4.Checked=true then begin
        nn:=DataModule5.kelimelerspam_frekans.AsInteger;
       DataModule5.kelimeler.Filter:='token='+QuotedStr(s);
       DataModule5.kelimeler.Filtered:=true;
        DataModule5.kelimeler.Edit;
        DataModule5.kelimelerspam_frekans.AsInteger:=nn+1;

DataModule5.kelimelermes_no.AsInteger:=DataModule5.mesajmesaj_no.AsInteger;
      DataModule5.kelimeler.Post;

        DataModule5.kelimeler.Filtered:=false;
       end ;
       if  CheckBox4.Checked=false then
         begin
          nn:=DataModule5.kelimelerham_frekans.AsInteger;
        DataModule5.kelimeler.Filter:='token='+QuotedStr(s);
        DataModule5.kelimeler.Filtered:=true;
          DataModule5.kelimeler.Edit;
         DataModule5.kelimelerham_frekans.AsInteger:=nn+1;

DataModule5.kelimelermes_no.AsInteger:=DataModule5.mesajmesaj_no.AsInteger;
          DataModule5.kelimeler.Post;

        DataModule5.kelimeler.Filtered:=false;
       end;

  end;

  end
else
if (length(s)>2) then begin
DataModule5.kelimeler.append;
DataModule5.kelimelertoken.AsString:=ansiLowerCase(s);
if CheckBox4.Checked=true then begin
DataModule5.kelimelerspam_frekans.AsInteger:=1;
DataModule5.kelimelermes_no.AsInteger:=DataModule5.mesajmesaj_no.AsInteger;
   end
 else
 begin
DataModule5.kelimelerham_frekans.AsInteger:=1;
DataModule5.kelimelermes_no.AsInteger:=DataModule5.mesajmesaj_no.asinteger;
 end;
DataModule5.kelimeler.Post;
end;
i:=y+1;
until (i>=length(f))
```

```
end;
adotable1.Next;
say:=say+1;
end;

  finally
    label1.visible:=true;
    label1.Caption:='';
    label1.Caption:='islem Tamamlanmistir!!!';

    Label7.Caption:='';
label7.Caption:=IntToStr(say)+'   adet mesaj islenmistir...!';
label8.Caption:='';
Label8.Caption:=IntToStr(DataModule5.mesaj.RecordCount);
ADOTable1.Close;
        end;
        end;

procedure TForm11.Button2Click(Sender: TObject);
var
z,i,bas,n,bit,p,x,y,cc,nn,st,say,ff:integer;
zz,t,h:double;
dn:byte;
f,s:string;
d:char;
begin
// burda

adotable1.Open;
adotable1.DisableControls;
adotable1.First;
DataModule5.kelimeler2.Close;
DataModule5.kelimeler2.Open;
adotable1.First;
say:=0;
while not adotable1.Eof do begin
ProgressBar1.StepBy(1);
Memo1.Text:='';
memo1.Lines.Add(adotable1icerik.AsVariant);
if CheckBox1.Checked=true then Memo1.Lines.Add(adotable1Konu.AsString);
if CheckBox2.Checked=true then Memo1.Lines.Add(adotable1kimden.AsString);
i:=0;
y:=0;
p:=0;
z:=0;

datamodule5.mesaj.Close;
datamodule5.mesaj.open;
```

```
datamodule5.mesaj.append;

if CheckBox4.Checked=true then DataModule5.mesajmesaj_sinifi.AsString:='spam'
else
DataModule5.mesajmesaj_sinifi.asstring:='normal';
datamodule5.mesaj.post;

for z:=0 to Memo1.Lines.Count-1 do begin
i:=0;
f:=Memo1.Lines[z];
if (length(f)=0) then continue;
repeat
x:=getXY(i,f);
if (x=length(f)) then break;
y:=getXY2(x+1,f);
s:=AnsiLowerCase(copy(f,x,y-x));
n:=0;
nn:=0;
DataModule5.kelimeler2.Locate('token',s,[]);
if DataModule5.kelimeler2token.AsString=s then
  begin
        if CheckBox4.Checked=true then begin
        nn:=DataModule5.kelimeler2spam_frekans.AsInteger;
        DataModule5.kelimeler2.Edit;
        DataModule5.kelimeler2spam_frekans.AsInteger:=nn+1;
        DataModule5.kelimeler2.Post;

       end ;
       if  CheckBox4.Checked=false then
         begin
          nn:=DataModule5.kelimeler2ham_frekans.AsInteger;
                DataModule5.kelimeler2.Edit;
          DataModule5.kelimeler2ham_frekans.AsInteger:=nn+1;
          DataModule5.kelimeler2.Post;
     end;
  end
else
if (length(s)>2) then begin
DataModule5.kelimeler2.append;
DataModule5.kelimeler2token.AsString:=ansiLowerCase(s);
if CheckBox4.Checked=true then
DataModule5.kelimeler2spam_frekans.AsInteger:=1 else
DataModule5.kelimeler2ham_frekans.AsInteger:=1;
DataModule5.kelimeler2.Post;
end;
i:=y+1;
until (i>=length(f))
end;
```

```
adotable1.Next;
say:=say+1;
end;
label1.visible:=true;
LABEL1.Caption:='';
label1.caption:='islem tamamlanmistir!!!';
Label7.Caption:='';
label7.Caption:=IntToStr(say)+'  adet mesaj islenmistir...!';
label8.Caption:='';
Label8.Caption:=IntToStr(DataModule5.mesaj.RecordCount);
ADOTable1.Close;
end;
procedure TForm11.FormShow(Sender: TObject);
begin
DataModule5.ADOConnection1.Close;
DataModule5.ADOConnection1.Open;
DataModule10.ADOConnection1.Close;
DataModule10.ADOConnection1.Open;
DataModule5.iletiler.Open;
DataModule5.kelimeler2.Close;
DataModule5.kelimeler2.Open;
DataModule5.mesaj.Close;
DataModule5.mesaj.Open;
label7.Caption:='';
label8.Caption:='';
label1.Visible:=false;
end;
procedure TForm11.Button3Click(Sender: TObject);
begin
form11.Close;
end;
procedure TForm11.Button4Click(Sender: TObject);
var
s:string;
begin
OpenDialog1.Execute;
s:=OpenDialog1.FileName;
ADOTable1.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;Data
Source='+s+';'+'Persist Security Info=False';
ADoTable1.Active:=true;
 ListBox1.Items.Add('***  '+s);
label5.Caption:='';
label5.Caption:=inttostr(adotable1.RecordCount);
ProgressBar1.Max:=adotable1.RecordCount;
ProgressBar1.Position:=0;
end;
end.
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Cüneyt Altunyaprak
Doğum Yeri : İSTANBUL
Doğum Tarihi : 1977
Medeni Hali : Bekâr

EĞİTİM VE AKADEMİK BİLGİLER

Lise : 1991-1994 Pertevniyal Lisesi/İSTANBUL
Lisans : 1995-2001 Muğla Üniversitesi Fen-Edebiyat Fakültesi İstatistik ve Bilgisayar Bilimleri
Yabancı Dili : İngilizce

MESLEKİ BİLGİLER

Mezuniyetimin ardından İstanbul'da bir bilgisayar eğitim ve danışmanlık firmasında sistem uzmanı ve eğitmen olarak görev yapan Cüneyt Altunyaprak, 2002 yılında Muğla Üniversitesi Bilgi İşlem Daire Başkanlığı'nda Uzman olarak göreve başlamıştır. 2003 yılında yüksek lisans programına başlamıştır. Halen Muğla Üniversitesi Bilgi İşlem Daire Başkanlığında Sistem ve Network Grubunda Sistem sorumlusu olarak görev yapmaktadır.